

LEVEL

AD A097417

Q

OSU

NUMERICAL ELECTROMAGNETIC CODE (NEC) - BASIC SCATTERING CODE

PART II: CODE MANUAL

The Ohio State University

F. W. Schmidt and R. J. Marhefka

AD A097417

The Ohio State University

ElectroScience Laboratory

Department of Electrical Engineering
Columbus, Ohio 43212

DTIC
ELECT
APR 7 1981

Technical Report 784508-14

September 1979

Best Available Copy

DTIC FILE COPY

Naval Regional Procurement Office
Long Beach, California 90822

Approved for public release;
distribution unlimited.

81 4

3 09

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Best Available Copy

**Best
Available
Copy**

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
	AD-A097 417	
4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED	
NUMERICAL ELECTROMAGNETIC CODE (NEC) - BASIC SCATTERING CODE - PART I: CODE MANUAL	2 Technical Report	
7. AUTHOR(s)	6. PERFORMING ORG. REPORT NUMBER	
F. W. Schmidt and R. J. Marhefka	ESI-784508-14	
	8. CONTRACT OR GRANT NUMBER(s)	
	15 Contract NO. 123-76-C-1371	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering Columbus, Ohio 43212		
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE	
Naval Regional Procurement Office Long Beach, California 90822	11 September 1979	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES	
	480	
	15. SECURITY CLASS. (of this report)	
	Unclassified	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
The material contained in this report is also used as a thesis submitted to the Department of Electrical Engineering, The Ohio State University as partial fulfillment for the degree Master of Science.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Geometrical Theory of Diffraction Uniform asymptotic solutions Plate models;		Cylinder model Far field pattern Computer code
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
The Numerical Electromagnetic Code - Basic Scattering Code is a user-oriented computed code for the analysis of the far field patterns of antennas in the presence of perfectly conducting metal structures at UHF and above. The analysis is based on uniform asymptotic techniques formulated in terms of the Geometrical Theory of Diffraction (GTD). Complicated structures can be simulated by arbitrarily oriented flat plates, an infinite ground plane, and a finite elliptic cylinder.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Best Available Copy

20.

A wide range of practical problems can be simulated using these shapes. For example, flat plates can be used to model the superstructure of a ship, the body of a truck, or the wings and stores of an aircraft. The finite elliptic cylinder can be used to model a mast or smoke stack of a ship, or the fuselage and engines of an aircraft.

This document describes the FORTRAN coding in detail. It gives background on practical aspects of the GTD and contains an overview of the code organization. This information will be of primary interest to someone attempting to modify the code. It will also be helpful when the code is being implemented on a computer system on which the coding may not be compatible.

TABLE OF CONTENTS

Chapter		Page
I	INTRODUCTION.....	1
II	BACKGROUND.....	3
	A. Qualitative Overview of GTD	
	B. Method Used in Computing the Fields Using GTD	
III	CODE ORGANIZATION.....	26
	A. Overview of Code	
	B. Subroutine and Common Block Linkages	
	C. Overlay Techniques	
	D. Dimensions for Sources, Plates, and Edges	
	E. Coordinate Systems	
IV	CODE DESCRIPTION.....	45
	A. Main Program	
	B. Subroutine Write-ups	
V	DEFINITION OF COMMON BLOCK VARIABLES.....	466
VI	SYSTEM LIBRARY FUNCTIONS USED BY CODE.....	478
	REFERENCES.....	479

Distribution/		
Availability Codes		
Initial Special		
A		

CHAPTER I INTRODUCTION

The Numerical Electromagnetic Code - Basic Scattering Code is a user-oriented computer code for the analysis of the far field patterns of antennas in the presence of perfectly conducting metal structures at UHF and above. Complicated structures can be simulated by arbitrarily oriented flat plates, an infinite ground plane, and a finite elliptic cylinder. This type of analysis has been used very successfully in the past to model aircraft shapes[1,2,3]. The present solution has been extended to include a wide range of problems. For example, flat plates can be used to model the superstructure of a ship, the body of a truck, or the wings and stores of an aircraft. The finite elliptic cylinder can be used to model a mast or smoke stack of a ship, or the fuselage and engines of an aircraft.

The analysis is based on uniform asymptotic techniques formulated in terms of the Geometrical Theory of Diffraction (GTD)[4,5,6]. The GTD approach is ideal for a general high frequency study of antennas in a complex environment in that only the most basic structural features of an otherwise very complicated structure need to be modeled. This is because ray optical techniques are used to determine components of the field incident on and diffracted by the various structures. Components of the diffracted fields are found using the GTD solutions in terms of the individual rays which are summed with the geometrical optics terms in the far field. The rays from a given scatterer tend to interact with other structures causing various higher-order terms. In this way one can trace out the various possible combinations of rays that interact between scatterers and determine and include only the dominant terms. Thus, one need only be concerned with the important scattering components and neglect all other higher-order terms. This method leads to accurate and efficient computer codes that can be systematically written and tested. Complex problems can be built up from simpler problems in manageable pieces.

The limitations associated with the computer code result from the basic nature of the analyses. The solution is derived using the GTD which is a high frequency approach. In terms of the scattering from plate structures this means that each plate should have edges at least a wavelength long. In terms of the cylinder structure its major and minor radii and length should be a wavelength in extent. In addition, each antenna element should be at least a wavelength from all edges and the curved surface. In many cases, the wavelength limit can be reduced to a quarter wavelength for engineering purposes.

Modeling small structures and antennas can be better accomplished using an integral equation solution such as NEC-Moment Methods[7]. The Basic Scattering Code has been interfaced with the Moment Method code so that the capabilities of both methods can be used to the fullest. For example, the Moment Method code can be used to analyze the currents and impedance of an antenna. The magnitude and phase of the current weights can then be used in the Basic Scattering Code to predict the far field patterns of the antennas in arbitrary pattern cuts.

There are two documents describing the NEC-Basic Scattering Code. Part I is a User's Manual[8] that contains a detailed description of the input parameters, an interpretation of the output, and example problems. The example problems are composed of sample input data with the resulting far field patterns compared against known results to confirm the validity of the code. Most users of the code will find that the User's Manual is sufficient to learn how to effectively operate the code.

This document is Part II. It describes the FORTRAN coding in detail. Chapter II gives background on practical aspects of the GTD. Several examples are shown to illustrate how the various GTD fields superimpose to give a total solution. Next, a particular GTD term is discussed in more detail to show the general concepts involved throughout the code. Chapter III contains an overview on how the code is organized. It describes the various coordinate systems involved, how a general subroutine is organized, and how the various subroutines are interrelated. Chapter IV contains for each subroutine: (1) a statement of purpose, (2) an illustration showing the geometry involved, (3) a brief narrative on the method used, (4) a flow diagram, (5) a dictionary of major variables, (6) a listing of the code. Chapter V defines the common blocks and Chapter VI lists the system library functions used by the code.

The information in the Code Manual will be of primary interest to someone attempting to modify the code. It will also be helpful when the code is being implemented on a computer system on which the coding may not be compatible.

CHAPTER II BACKGROUND

The Basic Scattering Code is used to evaluate the far field patterns of a given antenna in the presence of perfectly conducting scattering structures. It is a useful tool in the analysis and design of antenna placement and performance. This section provides the reader with background on how GTD is used in the code for computing the scattered fields. It also shows how to interpret and correlate the computed scattered fields to the specific geometry of a scattering structure. This chapter also provides a simple view of how the code generates a specific GTD scattered term. The explanations provided are general, giving an introduction to the more detailed descriptions provided later in the code manual. For a theoretical analysis of the methods used in the code, the reader is encouraged to refer to References 1, 4, 5, 6.

A. Qualitative Overview of GTD

The goal of the code is to solve for the fields scattered in a specified direction from a source (or set of sources) by the various features of a structure, as shown in Figure 1. The total field in a given observation direction is obtained by taking the sum of fields resulting from a number of different scattering mechanisms. Each component is determined by tracing the ray through the appropriate geometrical path and then using the Uniform Geometrical Theory of Diffraction to compute the magnitude and phase of the field if it has not been shadowed. The following examples serve to show the different mechanisms used in computing the scattered field and an example of typical fields resulting from such mechanisms.

Example consisting of a source and a single scattering element

The geometry used is a half-wave electric dipole mounted two wavelengths above a square plate four wavelengths on a side as shown in Figure 2. The total field of the source and structure is given by, $\bar{E} = \bar{E}^i + \bar{E}^s$ where $\bar{E}^i$ is the incident field:

$$\bar{E}^i = \begin{cases} \text{incident source field,} & \text{where ray is not shadowed} \\ 0 & \text{where source ray is shadowed,} \end{cases}$$

and $\bar{E}^s$ is the GTD scattered field:

$$\bar{E}^s = \begin{cases} \text{scattered fields,} & \text{where the rays are not shadowed,} \\ 0 & \text{where the rays are shadowed.} \end{cases}$$

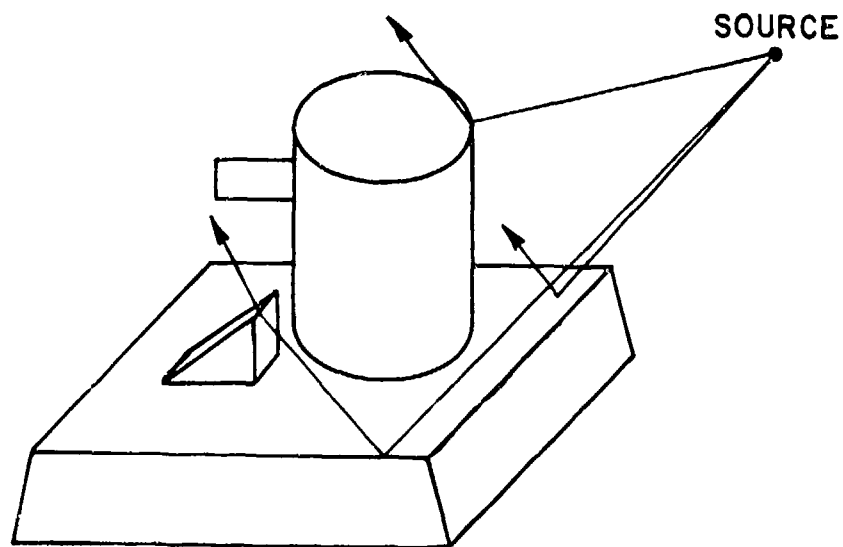


Figure 1--Illustration of general GTD problem.

The GTD scattered field is composed of the reflected fields, diffracted fields, etc. The source and reflected fields comprise the geometrical optics fields (G.O.).

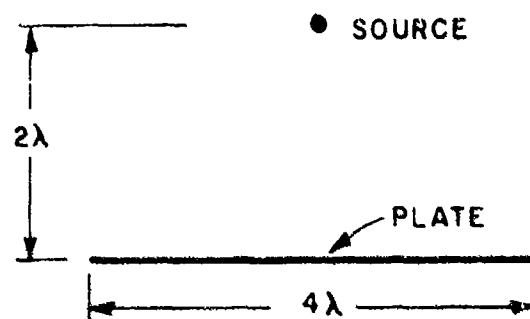


Figure 2--Geometry for a source in the presence of a plate.

Several single-order terms are used to compute the fields (in the far-zone) scattered by this structure. The word "order" here refers to the number of times the particular scattering term interacts with the body as it propagates from the source to the observation point. The source (or incident) field is that field which propagates straight from the source into the far field in the direction of the observer as shown in Figure 3. The pattern of the source field, in the presence of the plate, taken in the plane of the page is shown in Figure 4. The scale used in the patterns of Figures 4-10 is 0 to -40 dB. They are normalized to the maximum value of the total field pattern in Figure 10.

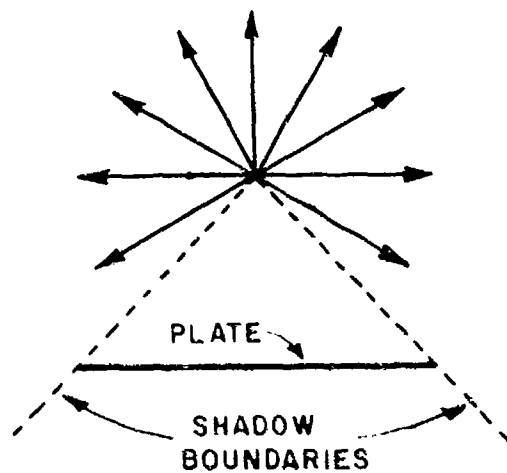


Figure 3--Illustration of source ray paths.

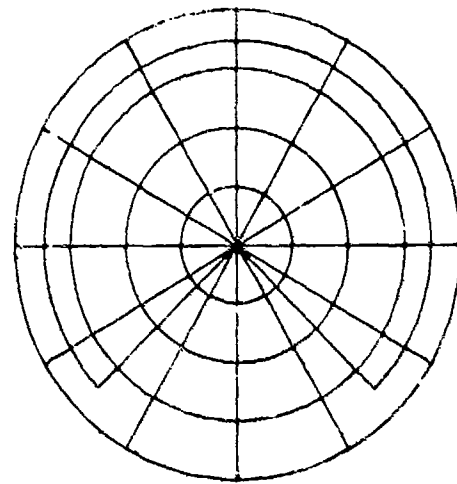


Figure 4--Source field.

The reflected field is simply the geometrical optics field reflected by the plate as shown in Figure 5. The fields due to the reflection mechanism, shown in Figure 6, are easily obtained from image theory. The total geometrical optics fields (the sum of the direct and reflected fields) are plotted in Figure 7.

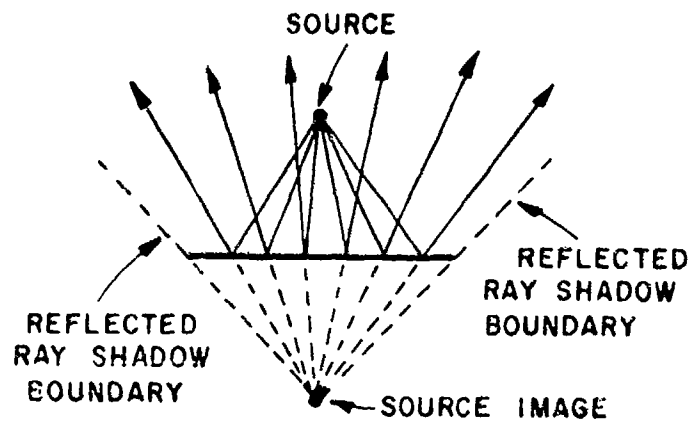


Figure 5--Illustration of plate reflected ray paths.

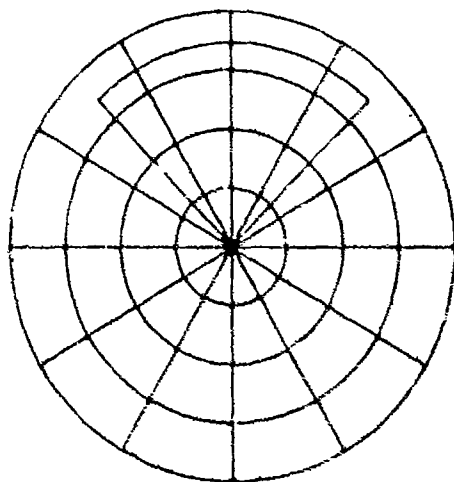


Figure 6--Field reflected from plate.

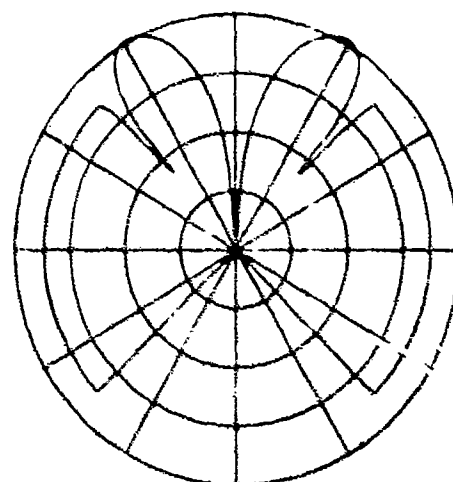


Figure 7--Geometrical optics field which is the sum of the incident and reflected fields.

The diffracted fields, which include edge, slope, and corner diffraction, are shown in Figure 8. The ray paths for edge diffraction are shown in Figure 9. Figure 10 shows the total scattered field. Note that the diffracted field smooths out the discontinuities on the G.O. fields. Although the diffracted field magnitude is continuous at the shadow boundary, the phase jumps by 180° there. This subtracts from the lit side and adds to the shadow side, smoothing out the discontinuity. Higher order terms (such as double diffraction) could also be computed to further improve the accuracy of the solution.

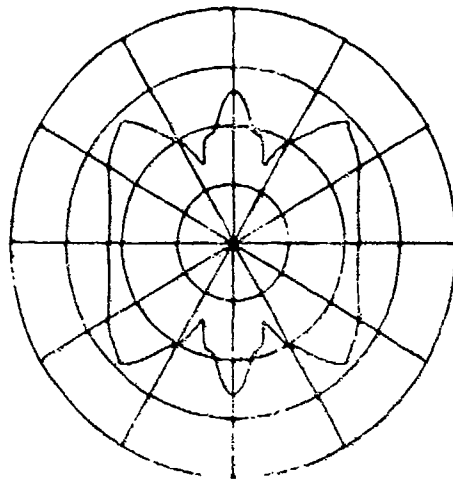


Figure 3--Diffracted fields.

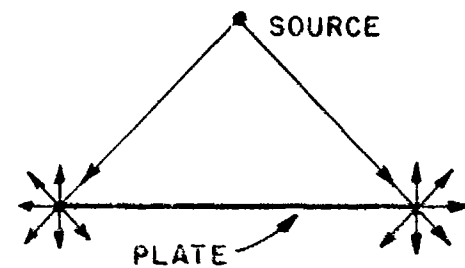


Figure 9--Illustration of diffracted rays.

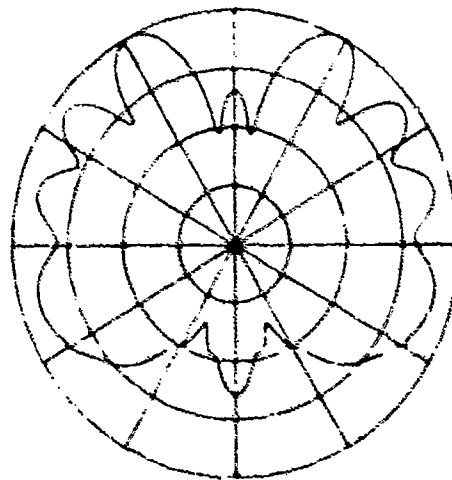


Figure 10--Total pattern.

Example consisting of a source and three scattering elements

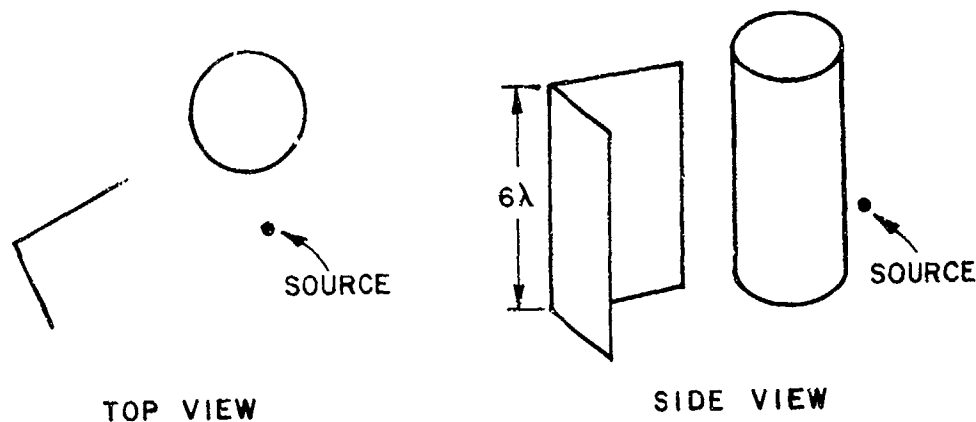


Figure 11--Illustration of source and scattering elements.

The geometry used for this example is shown in Figure 11. As with the previous example, the source field, single order plate reflections, and diffractions exist, as is shown in Figures 12-16. Field patterns in Figures 11-36 are taken in the plane normal to the cylinder and plotted with a scale from 0 to -40 dB such that they are normalized to the maximum value of the total field pattern in Figure 36.

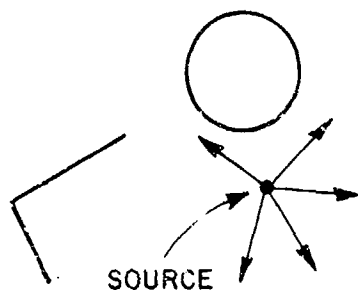


Figure 12--Source field ray paths.

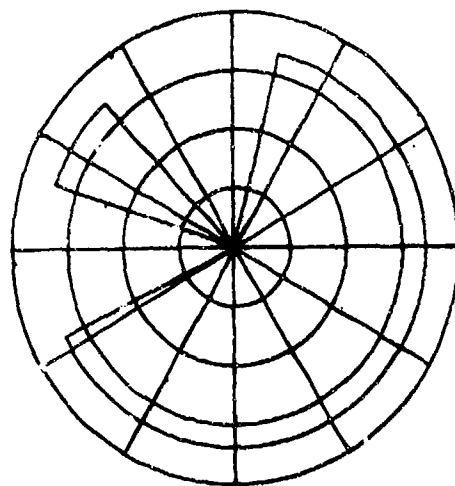


Figure 13--Source fields.

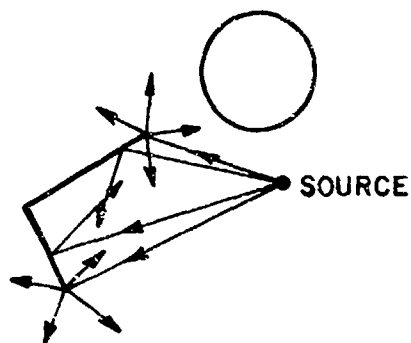


Figure 14--Illustration of first order plate ray paths.

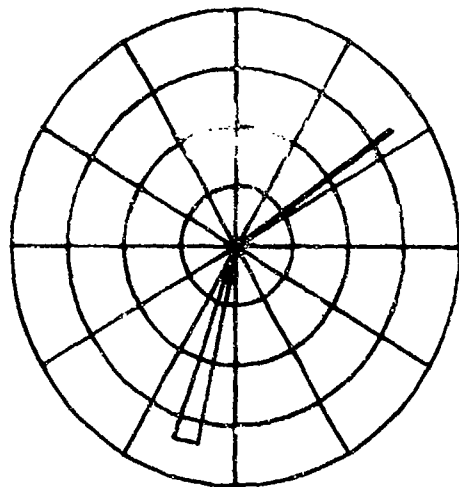


Figure 15--Fields due to single order plate reflection.

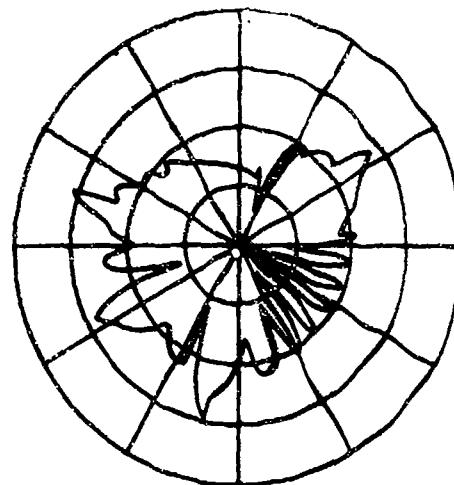


Figure 16--Fields due to plate diffraction.

In addition to first order plate terms there are first order cylinder terms: 1) the scattered (reflected and diffracted) fields from the cylinder's curved surface, 2) the field reflected from the end caps and 3) the fields diffracted by the end cap rims. These are shown in Figures 17-21. Note that in the geometry presented in Figure 11, end cap reflections will not occur. Therefore, a different geometry is shown in Figure 21 to demonstrate end cap reflections. Note that with more than one body present, individual terms are often shadowed by other bodies in the structure, creating

discontinuities as shown in many of the figures (as in Figure 18 for the cylinder scattered fields).

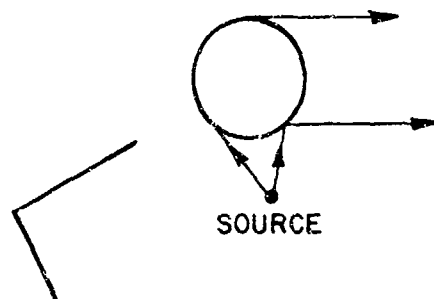


Figure 17--First order ray paths for the cylinder's curved surface.

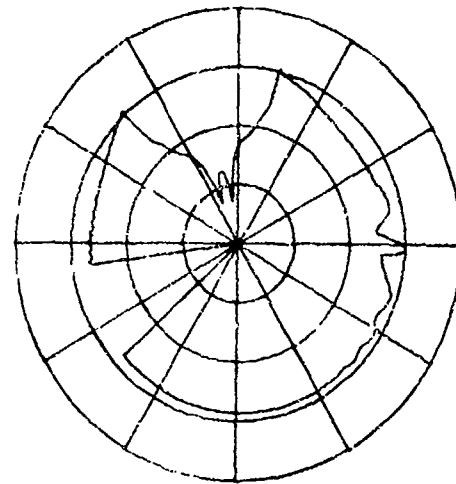


Figure 18--First order cylinder curved surface scattered field.

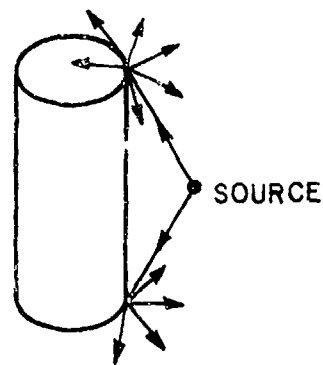


Figure 19--Ray paths for end cap diffracted fields.

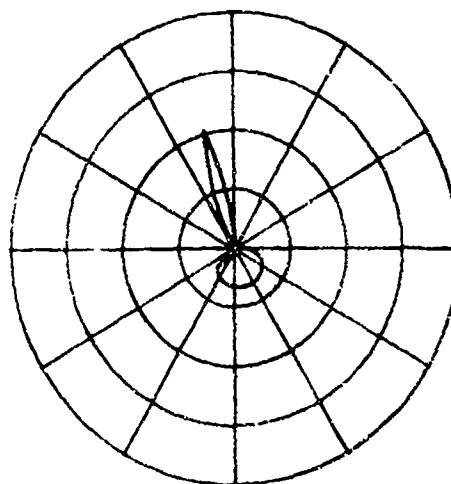


Figure 20--Fields due to end cap diffraction.

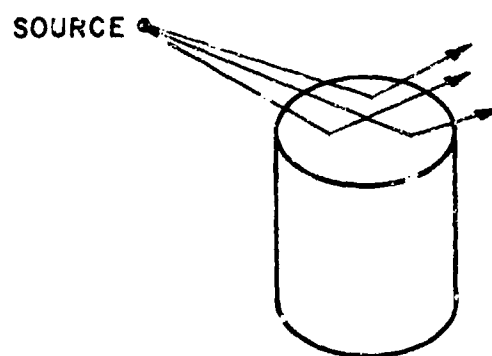


Figure 21--Illustration of ray paths for end cap reflected fields.

In addition to single order mechanisms, second order scattering occurs where the ray is scattered by one body and then scattered by the second. Several different double scattering (or second order) terms are computed. Double reflection, where a ray is reflected by one plate and then by another, is shown in Figures 22 and 23.

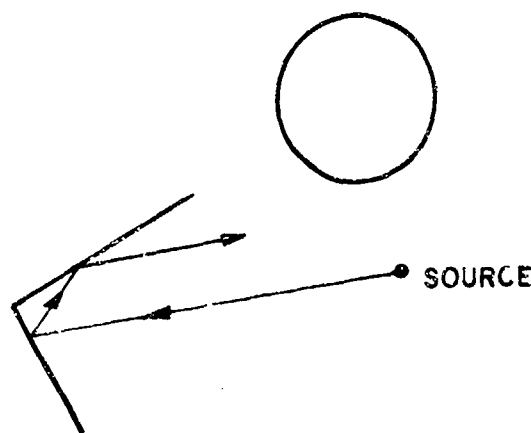


Figure 22--Ray path for double reflected fields.

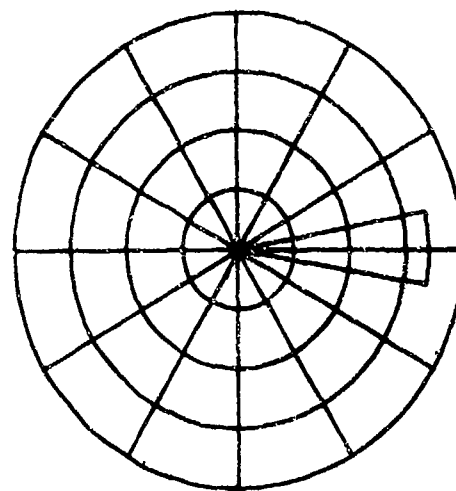


Figure 23--Fields due to double reflected rays.

Another second order scattering mechanism involving plates is reflection-diffraction, where a ray is reflected from one plate and diffracted by another. This is illustrated in Figures 24 and 25. The inverse mechanism, diffraction-reflection, illustrated in Figures 26 and 27, involves fields diffracted from a plate edge and then reflected by another plate.

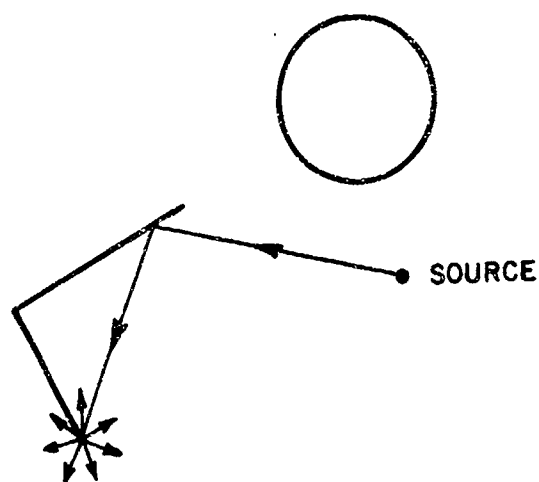


Figure 24--Ray paths for plate reflection-diffraction.

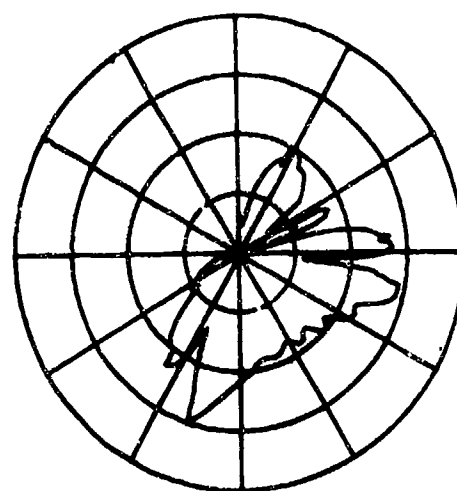


Figure 25--Fields resulting from plate reflection-diffraction.

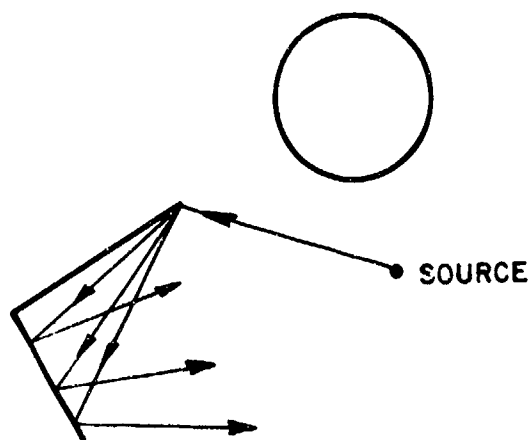


Figure 26--Illustration of plate diffracted-reflected ray paths.

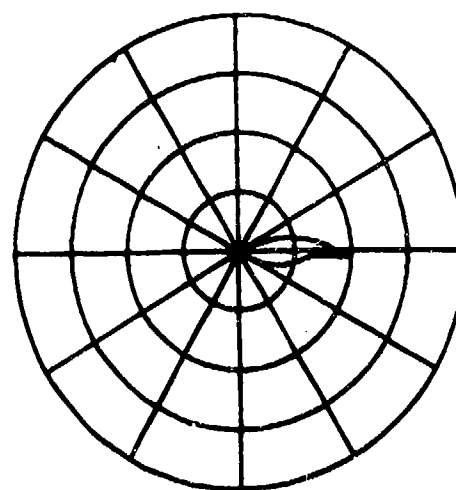


Figure 27--Fields due to plate diffraction-reflection mechanism.

A number of the scattering mechanisms involve interactions between the cylinders and one of the plates. Two such terms result from scattering of the fields by the cylinder and then reflection by a plate and vice-versa. Figures 28 and 29 illustrate the ray paths and fields of rays which are reflected from a plate and then scattered by the elliptic cylinder. Figures 30 and 31 illustrate ray paths and fields resulting from ray scattered by the cylinder and then reflected by a plate.

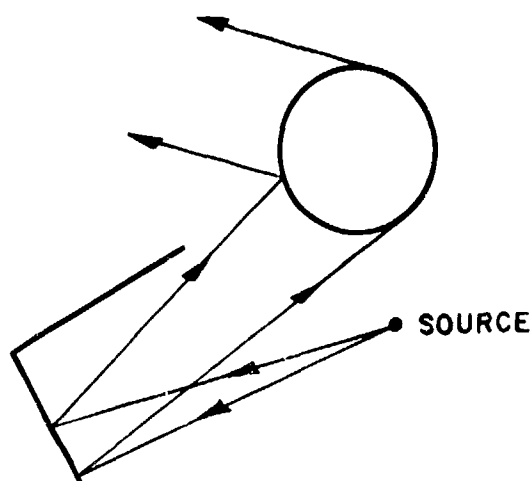


Figure 28--Illustration of rays reflected by a plate and scattered by the cylinder.

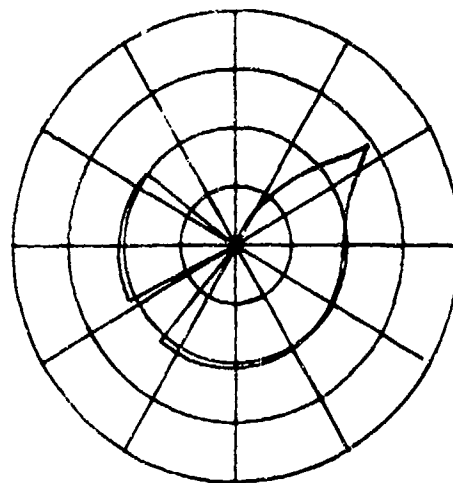


Figure 29--Fields reflected by plates and then scattered by the cylinder.

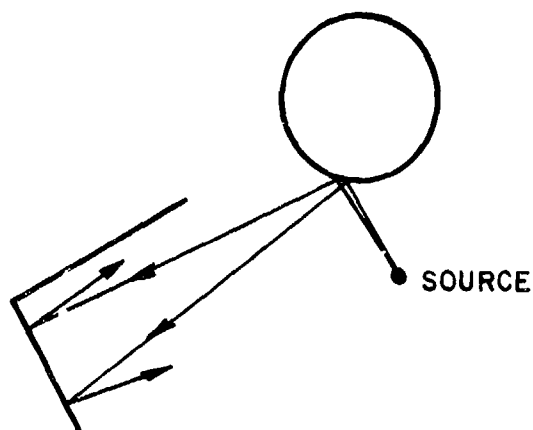


Figure 30--Illustration of rays scattered by cylinder and then reflected by a plate.

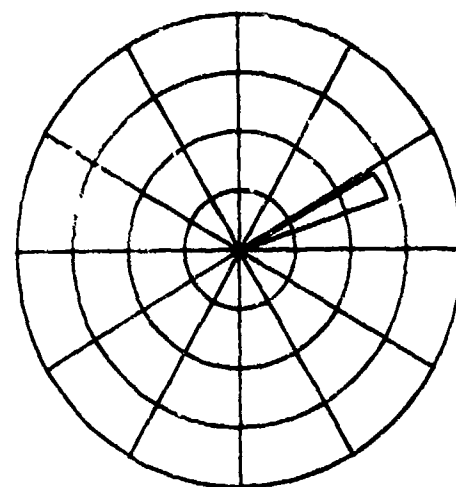


Figure 31--Fields scattered by the cylinder and reflected by a plate.

Another second order scattering mechanism involves fields reflected by the cylinder and then diffracted by a plate edge. The ray paths and fields for this term are illustrated in Figures 32 and 33. The inverse of this term is the fields of rays diffracted by a plate edge and then reflected by the cylinder, as shown in Figures 34 and 35.

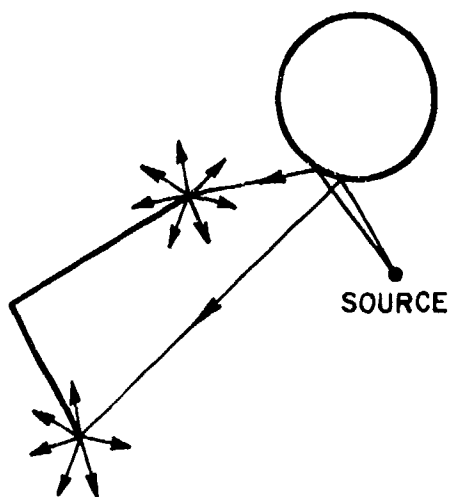


Figure 32--Illustration of ray reflected by cylinder and diffracted by plate edge.

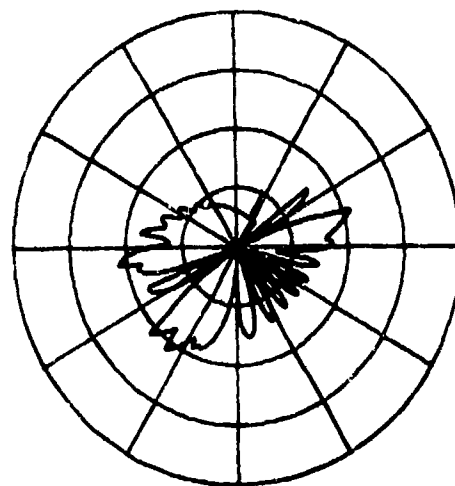


Figure 33--Fields reflected by cylinder, diffracted by plate edges.

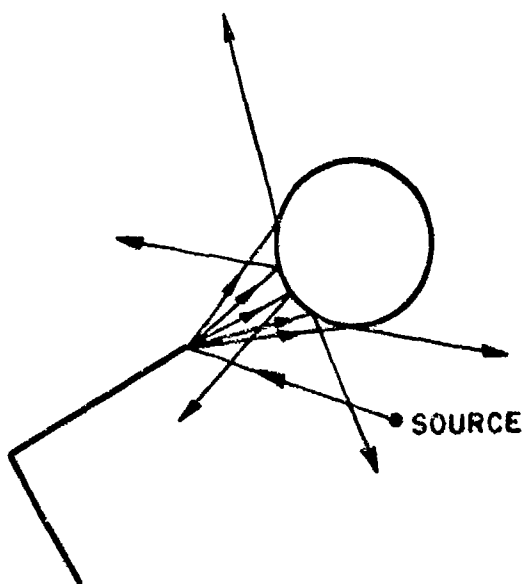


Figure 34--Illustration of rays diffracted by plate edge and reflected by cylinder.

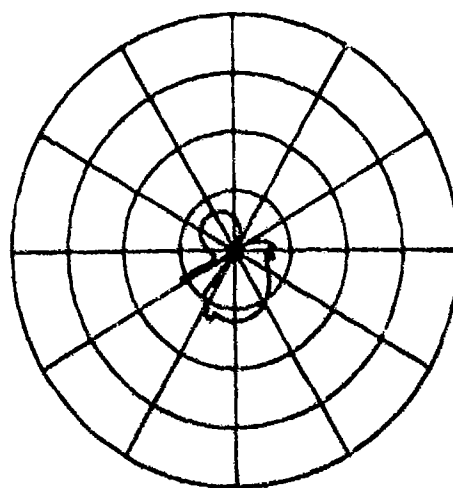


Figure 35--Fields diffracted by plate, reflected by cylinder.

The total pattern is obtained by summing the field components for the mechanisms mentioned previously. The total field pattern is illustrated in Figure 36.

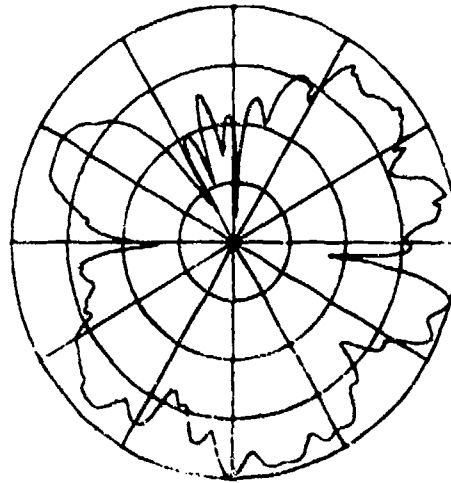


Figure 36--Total fields of source in the presence of scattering bodies.

Higher order scattering terms can also be computed, which will in some cases improve the accuracy of the field computations. Generally, it is found that such terms are negligible in magnitude as well as being difficult to compute and therefore are not included in the code. The presence of discontinuities in a final field pattern, however, indicates the presence of regions where higher-order terms are needed.

B. Method Used in Computing the Fields Using GTD

In order to use the Basic Scattering Code, the user first specifies a set of observation angles, for which he desires to obtain the far field pattern of the source(s) in the presence of a structure. The code computes the fields over the pattern angles specified for each source defined and uses superposition to obtain the total fields. For each observation direction computed, the code computes every GTD term applicable to the structure at hand, unless the user limits the types of terms computed. Each term is computed independently of the others. The following gives an outline of the procedure used in computing a particular GTD term.

The code first analyzes the input geometry in the geometry subroutines. Many of the parameters which do not vary for a given

geometry are computed there in advance. This avoids re-computation of fixed variables. It also gives an a priori indication of the regions in which different GTD terms need to be included. This allows the code to avoid performing computations where not necessary.

Two examples of GTD problems involving first and second order scattering phenomena are shown in Figures 37 and 38, respectively. A basic outline of how the various fields are computed is as follows:

1. Make any a priori checks of the fixed geometry
2. Compute ray path for specific mechanism desired
3. Determine if ray is blocked anywhere by another part of the structure
4. Use theory to compute the magnitude and phase of the field component resulting from the mechanism. If a second order mechanism is involved this is a two step process where the field of the first interaction is computed and then the field of the second interaction is computed.

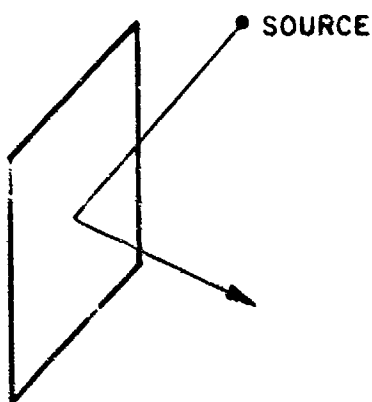


Figure 37--First order scattered term.

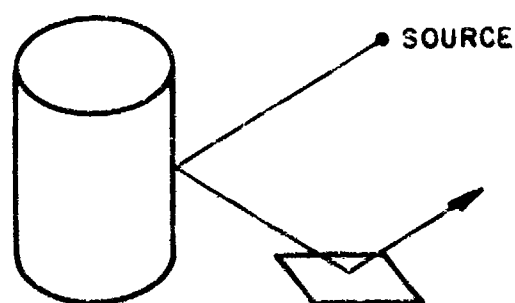


Figure 38--Second order scattered term.

The following is a more specific example of how the code computes the fields of a second order scattering term. The geometry, consisting of a source and four plates, is illustrated in Figure 39.

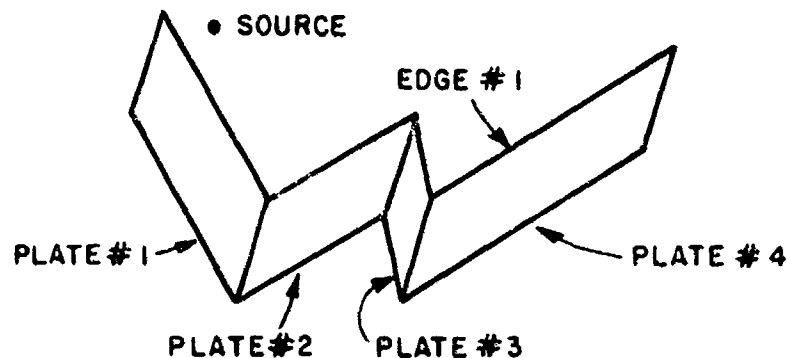


Figure 39--Illustration of a multiple plate example.

The code starts by choosing a source and a particular field term. As an example, let us choose the field reflected by plate #1 and then diffracted by edge #1 of plate #4. The code next chooses an observation angle and performs the following tasks:

- 1) The fixed geometry bounds are checked to see if a diffraction can occur in the direction specified. If it can, the code proceeds to the next step. If it can't, the code sets the fields to zero.
- 2) Determination of ray path. The code establishes the ray path, which includes both the reflection and diffraction points, as well as the propagation direction of the ray. It is temporarily assumed that plate #1 is of infinite extent, and that no shadowing occurs. It is also assumed that edge #1 of plate #4 is infinite. This guarantees that the reflection-diffraction will occur. The plate reflection can be handled by using the image of the source in plate #1 and removing the plate, as all the plate reflected rays appear to emanate from the image location (as shown in Figure 40).

Once the diffraction point is known, the reflection point on plate #1 is found by determining the line from the source image to the diffraction point. This reflection point is the intersection between this line and plate #1 as shown in Figure 42.

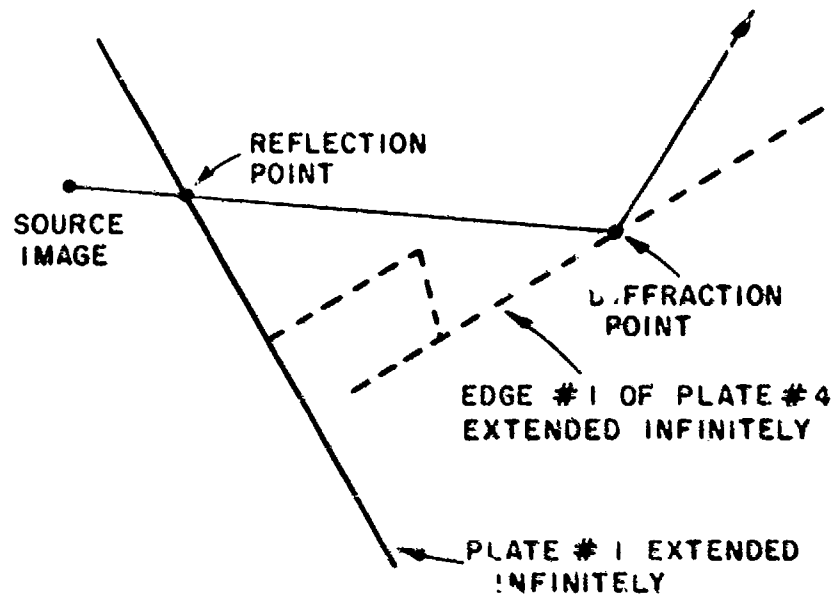


Figure 42--Geometry used in finding reflection point on plate #1.

The code then determines if the ray path is valid for the finite geometry of the structure. The reflection point is valid if the line drawn from the image source to the diffraction point passes through the finite plate (plate #1) as shown in Figure 43. The diffraction point is valid if the point lies along the limits of the finite edge (see Figure 44). Figure 45 shows the computed ray path (assuming the scatterer points do lie on the finite plates).

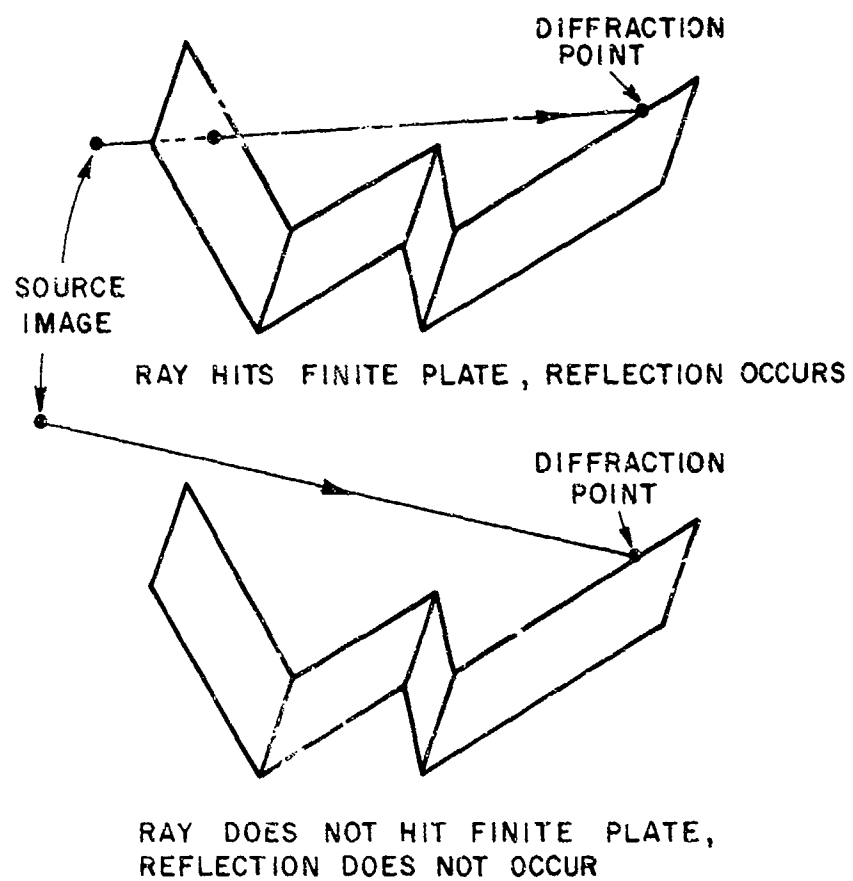


Figure 43--Illustration of test for reflections from plate #1 for two different cases.

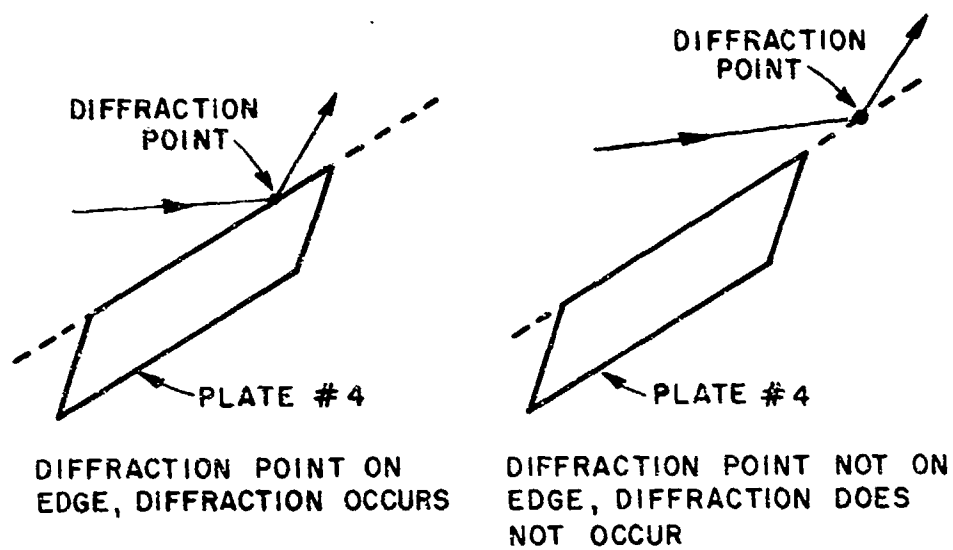


Figure 44--Illustration of test for diffraction from edge #1 of plate #4.

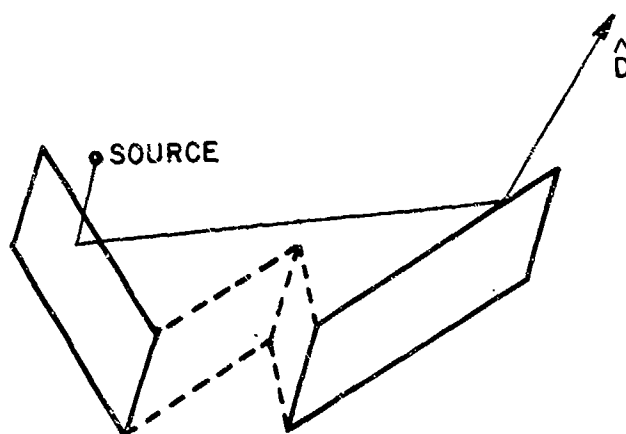
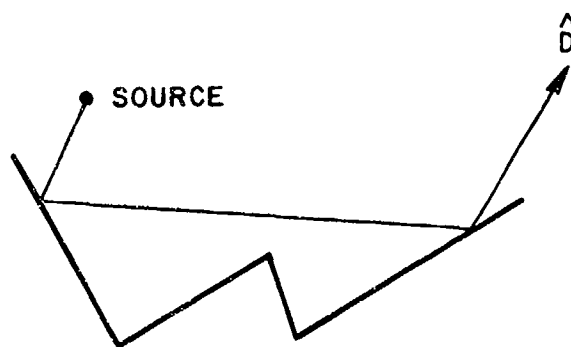


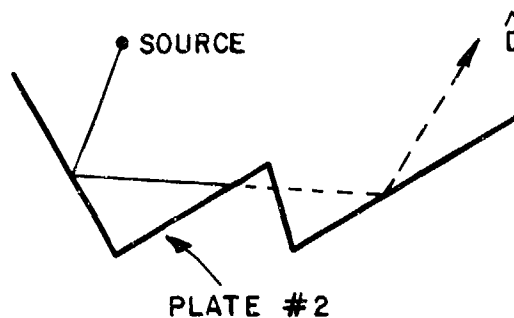
Figure 45--Illustration of the ray path for a field reflected from plate #1 then diffracted from edge #1 of plate #4.

3. Test for ray shadowing

The code next checks to see if the ray is obstructed anywhere along its path. The code first checks to see if the ray is shadowed after the diffraction by determining if a ray traveling in direction $\hat{D}$ from the diffraction point will hit a plate or a cylinder. The code then checks to see if the ray is shadowed between the reflection and diffraction points. In the example given this is the critical area, as there are other plates in this vicinity (see Figure 46).



Ray not shadowed between reflection and diffraction points.



Ray shadowed by plate #2.

Figure 46--Illustration of test for the shadowing of the ray of interest by a plate.

The code then checks to see if the source ray is shadowed by a plate or cylinder. If it is found that the ray is shadowed or that the reflection diffraction specified cannot happen for the geometry at hand, the code suspends computation of the term and sets the reflected-diffracted ray's field to zero.

4a. Computation of reflected field incident on the diffraction point.

The reflected field is simply obtained by using the image source located at the image position and computing the "free-space" fields incident on the diffraction point.

4b. Computation of the diffracted field :

The diffracted field is obtained by multiplying the field incident on the edge by the edge diffraction coefficients. The parameters needed for the diffraction coefficients are obtained from the geometry of the incident and diffracted rays and the edge. The phase of the diffracted ray is then referred to the coordinate system origin and the task is completed.

CHAPTER III CODE ORGANIZATION

The information in Chapter II is designed to present a qualitative view as to how the GTD is systematically used to construct a solution to a couple of problems. This chapter is intended to present how specific pieces of the code relate to the computation of the scattered fields.

First a brief outline is given in Section A. In Section B, tables are given, showing the interrelationships of the subroutines and the common blocks. Ways of reorganizing the code into smaller pieces are discussed in Section C. In Section D, a description of which variables must be reimensioned to change the maximum number of sources, plates, and edges is given.

In the last section, a brief discussion of most of the coordinate systems used in the code is presented. This is intended to provide a convenient way of reducing repetitive discussion of these systems throughout the subroutine descriptions.

A. Overview of Code

The Basic Scattering Code is organized in a systematic way to increase the efficiency of computation by reducing core swapping and to allow different pieces to be run separately. This feature can be quite useful when it is necessary to run the code in a limited amount of core.

The various operations of the code are carried out in different classes of subroutines such as field computation, geometry, shadowing, ray tracing, and other service subroutines. Many of the subroutines are classified along with a brief description of their principal functions in Table 1.

The MAIN program provides the overall control of the various operations of the code. It controls the input of the geometry data, which is described in detail in the User's Manual [8]. It prepares the data for computation by transforming the input data into the optimum coordinate system for computations and by normalizing variables into common units of wavelengths. It also directs the computation of the various GTD fields and superimposes these fields to obtain the total far field pattern. The overall structure of the computation section of the code is outlined below.

The main computation section is composed of a number of large DO loops that step through all the various sources, GTD field types, scattering centers, and observation directions. Each loop ends at a common point where the different fields are superimposed and stored.

The first loop steps through the different sources. For a particular source the fixed properties of the geometry of the problem are first determined and stored. This includes the a priori bounds on the diffracted fields. These parameters are calculated in the geometry subroutines GEOM, GEOMC, and GEOMPC.

The MAIN program then loops through the various types of GTD fields which are identified by integers K and J as shown in Table 1. K=1 corresponds to fields involving plates, K=2 corresponds to cylinder fields, and K=3 to plate cylinder interaction fields. If only plates are present only the K=1 subroutines are called, if only a cylinder is present, only K=2 subroutines are called. If both plates and cylinders are present in the geometry, all three groups of subroutines (K=1,2,3) are called.

The MAIN program then loops through the various pattern angles desired. The observation directions are defined in the pattern cut coordinate system, discussed in Section III-E-4. Subroutine PATROT converts the observation direction to the reference coordinate system (RCS) discussed in Section III-E-1.

The MAIN program now branches to the section of the code where the specified GTD field subroutine is to be called. In this area the code loops through every combination of plates, edges, endcaps, etc. which apply to the GTD field (based on K and J) being computed. For example, if the fields reflected by one plate and then diffracted by another (K=1, J=5) are being computed, this section specifies every plate-edge combination in the geometry. This operation varies with the specific term being computed. Details are given in the MAIN program write up on K-J sections in Chapter IV-A. For each combination of plates, edges, etc., the MAIN program calls the appropriate field computation subroutine (listed in Table 1 according to K and J) to compute the scattered field.

This above arrangement of DO-loops is illustrated in the following flow diagram.

FLOW DIAGRAM

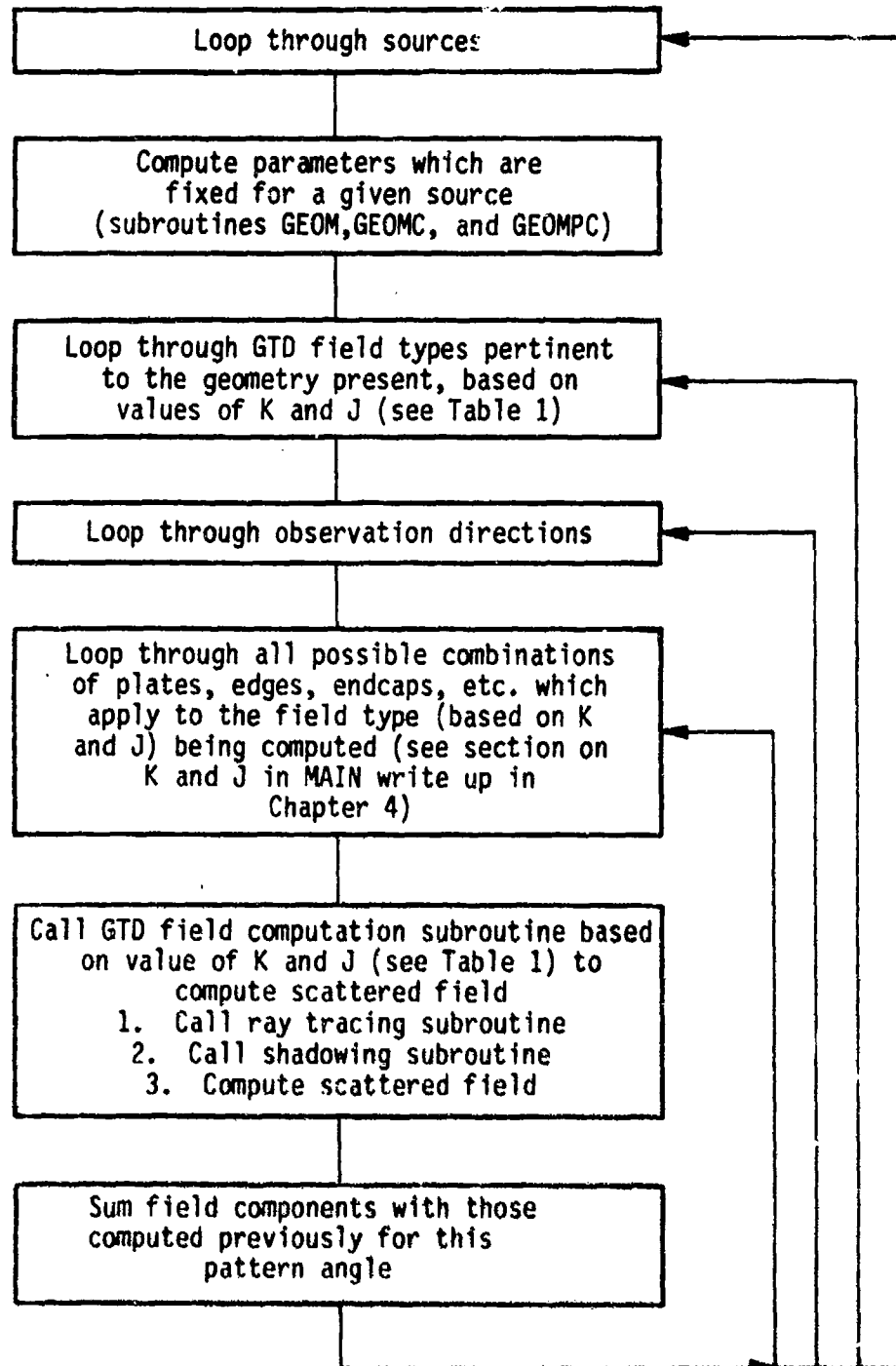


TABLE 1
LIST OF SOME IMPORTANT SUBROUTINES AND THEIR FUNCTION

Plate Field Subroutines K=1

J=1 INCFLD - direct field
J=2 REFPLA - field reflected from a plate
J=3 RPLRPL - field doubly reflected by plates
J=4 DIFPLT - field diffracted by a plate
J=5 RPLDPL - field reflected by a plate then diffracted by a plate
J=6 DPLRPL - field diffracted by a plate then reflected by a plate

Cylinder Field Subroutines K=2

J=1 SCTCYL - field scattered by a cylinder
J=2 REFCAP - field reflected by an end cap
J=3 ENDIF - field diffracted by an end cap rim

Plate-Cylinder Interaction Field Subroutines K=3

J=1 RPLSCL - field reflected by a plate then scattered by a cylinder
J=2 SCLRPL - field scattered by a cylinder then reflected by a plate
J=3 RCLDPL - field reflected by a cylinder then diffracted by a plate
J=4 DPLRCL - field diffracted by a plate then reflected by a cylinder.

Geometry Subroutines

GEOM - fixed geometry of the plates
GEOMC - fixed geometry of the cylinder
GEOMPC - fixed geometry of the plate-cylinder interactions.

Shadowing Subroutines

PLAINT - shadowing due to plates
CYLINT - shadowing due to the cylinder
CAPINT - shadowing due to the end caps

Ray tracing Subroutine

DFPTCL - diffraction points on end cap rims
DPTNFW - near field diffraction point on plate edge
DFPTWD - far field diffraction point on plate edge
DFRFPT - diffraction point on plate edge then reflection point on cylinder
RDFDPT - reflection point on cylinder then diffraction point on plate edge
RFPTCL - far field reflection point on cylinder
RDFDIN - near field reflection point on cylinder

The computation loops are nested in this manner, because for a given source and GTD field, a minimal number of subroutines need to be present in the computer core and they can stay there for a longer time than for any other loop configuration.

The field subroutines, whether they deal with reflection or diffraction for a plate or a cylinder, all have the same basic construction. The field subroutines start by checking the a priori bounds for the GTD mechanism of interest to see if it can produce a field propagating in the given observation direction. If it can, the code proceeds to trace the path back from the observation direction to the scattering points to the source without regard to the other structures in the geometry. This is accomplished in the ray tracing subroutines listed in Table 1. After the ray path is found, the path is tested to see if it is shadowed by any of the structures in the geometry. The shadowing subroutines are listed in Table 1. If the ray path has passed all the above tests the actual field calculation begins. First the field incident on the scattering point(s) is computed. The polarization is then converted to the proper canonical coordinate system for the particular GTD field. These coordinate systems are briefly discussed in section III-E. The use of the canonical system greatly simplifies the computation of the fields. Next, the reflection and/or the diffraction coefficients are computed for the problem. For example, the diffraction coefficient of the edge is computed in subroutine DW and its associated subroutines. The incident field is then multiplied by the reflection and/or diffraction coefficients along with the spread factors to compute the GTD field. The field polarization is then converted back to the reference coordinate system and the far field phase factor is added. This phase factor is the usual one given by

$$\frac{e^{-jks}}{s} = e^{jk\vec{X} \cdot \hat{D}} \frac{e^{-jkR}}{R}$$

where s is the radial distance out from the scattering point, $\vec{X}$ is the location of the scattering point, $\hat{D}$ is the radial vector pointing in the observation direction, and R is the radial distance out from the reference coordinate system origin in the far field observation direction. This is illustrated in Figure 47. The factor e^{-jkR}/R is suppressed at this point of the computations. It is added later in subroutine OUTPUT, for display purposes only, if the user specifies the distance R . The source weight, W , is also not added at this point in the calculations for convenience. The field therefore has the form

$$\vec{E} = W(\vec{E}_\theta + \vec{E}_\phi) \frac{e^{-jkR}}{R}$$

where $\vec{E}_\theta$ and $\vec{E}_\phi$ are calculated in the field computation subroutines.

8. Subroutine and Common Block Linkage

In addition to the subroutines that have been classified in Table 1, there are many important subroutines that provide necessary services to other subroutines and to the code in general. The linkage of these is quite complicated because of the interdependence of many of the subroutines. A list of all the subroutines and the subroutines in which they are used is given in Table 2.

The majority of the data information transferred in the code is done by named common blocks. This method provides the most efficient and direct scheme for the large amounts of information present through the intertwined subroutines of the code. A list of all the COMMON BLOCKS and the subroutines which use them is given in Table 3.

TABLE 2
LINKAGE BETWEEN SUBROUTINES

SUBROUTINE	SUBROUTINES IN WHICH IT IS USED
MAIN	Not applicable
BABS	MAIN, DFPTCL, DI, OUTPUT, POLYRT, PRIOUT
BLOCK DATA	Not applicable
BLOG10	OUTPUT
STAN2	MAIN, CAPINT, CYLINT, DIFPLT, DPLRCL, DPLRPL, ENDIF, GEOM, GEOMPC, OUTPUT, PATROT, PLAIN, PRIOUT, RCLDPL, RCLRPL, REF8P, REFCYL, RFDFIN, RFDFPT, RFPTCL, RPLDPL, RPLRCL, RPLSCL, SCLRPL, SCTCYL, SOURCE, TANG
CAPINT	CYLINT, GEOM, REFCAP
CYLINT	DIFPLT, DPLRCL, DPLRPL, GEOM, INCFLD, RCLDPL, RCLRPL, REFPLA, RPLDPL, RPLRPL, SCLRPL
DFPTCL	ENDIF
DFPTWO	DIFPLT, DPLRPL, RPLDPL
DFRPT	DPLRCL
DI	DIFPLT, DPLRPL, DW, RPLDPL
DIFPLT	MAIN
DPI	DW
DPLRCL	MAIN
DPLRPL	MAIN
OPTNFW	GEOMPC
DQG3?	FKARG, RPLSCL, SCLRPL, SCTCYL
DW	DIFPLT, DPLRCL, DPLRPL, RCLDPL, RPLDPL
DZ	ENDIF
ENDIF	MAIN
FCT	RPLSCL, SCLRPL, SCTCYL
FFCT	DI, DIFPLT, DPLRPL, RPLDPL
FKARG	RPLSCL, SCLRPL, SCTCYL
FKY	DZ
FRNELS	DI, DPI, FFCT, FKY, RPLSCL, SCLRPL, SCTCYL
FUNT	FKARG
GEOM	MAIN
GEOMC	MAIN
GEOMPC	MAIN
IMAGE	DPLRPL, GEOM, RCLRPL, SCLRPL
IMCDIR	GEOMC
IMDIR	GEOM, RPLRPL
INCFLD	MAIN
NANDB	DPLRCL, ENDIF, RCLDPL, RCLRPL, REFCYL, RPLRCL, RPLSCL, SCLRPL, SCTCYL
OUTPUT	MAIN
PATROT	MAIN
PFUN	RPLSCL, SCLRPL, SCTCYL
PLAIN	DIFPLT, DPLRCL, DPLRPL, ENDIF, GEOM, INCFLD, RCLDPL, RCLRPL, REFCAP, REFCYL, REFPLA, RPLDPL, RPLRCL, RPLRPL, RPLSCL, SCLRPL, SCTCYL
POLYRT	DFPTCL, RFDFIN
PRIOUT	MAIN
QFUN	RPLSCL, SCLRPL, SCTCYL
RAOCV	RPLSCL, SCLRPL, SCTCYL
RCLDPL	MAIN
RCLRPL	SCLRPL
REF8P	DPLRPL, RCLRPL, REFPLA, RPLDPL, RPLRCL, RPLRPL, RPLSCL, SCLRPL
REFCAP	MAIN
REFCYL	SCTCYL
REFPLA	MAIN
RFDFIN	GEOMPC
RFDFPT	RCLDPL
RFPTCL	RCLRPL, REFCYL, RPLRCL
ROTR4	MAIN
RPLDPL	MAIN
RPLRCL	RPLSCL
RPLRPL	MAIN
RPLSCL	MAIN
SCLRPL	MAIN
SCTCYL	MAIN
SOURCE	DIFPLT, DPLRCL, DPLRPL, ENDIF, INCFLD, RCLDPL, RCLRPL, REFCAP, REFCYL, REFPLA, RPLDPL, RPLRCL, RPLRPL, RPLSCL, SCLRPL, SCTCYL
SOURCEP	DIFPLT, DPLRPL, RPLDPL
TANG	CYLINT, GEOMC, GEOMPC

TABLE 3
LINKAGE BETWEEN COMMON BLOCKS AND SUBROUTINES

COMMON BLOCK	SUBROUTINES IN WHICH IT IS USED
BNDCL	DFRFT, DPLRCL, GEOMPC
BNDCL	DIFPLT, DPLRCL, DPLRPL, GEOM, GEOMPC
BNDICL	GEOMPC, RFPTCL, RPLRCL, RPLSCL
BNDRCL	GEOMPC, RCLDPL, RFDFPT
BNDSCCL	CYLINT, GEOMC, GEOMPC, RCLRPL, REFCYL, RFPTCL, SCLRPL, SCTCYL
BRNPHW	DFRFT, GEOMPC, RFDFPT
CLDRCL	MAIN, DPLRCL
CLRDC	MAIN, RCLDPL
CLRFC	MAIN, REFCYL
CLRFI	MAIN, RPLRCL
CLRFS	MAIN, RCLRPL
COMP	MAIN, BLOCK DATA, ENDIF, INCFLD, RPLSCL, SCLRPL, SCTCYL
DIP	MAIN, DFPTCL, DFRFT, DIFPLT, DPLRCL, DPLRPL, ENDIF, INCFLD, RCLDPL, RCLRPL, REFCAP, REFCYL, REFPLA, RFDFPT, RPLDPL, RPLRCL, RPLRPL, RPLSCL, SCLRPL, SCTCYL
DOUBLE	MAIN, DIFPLT
EDMAG	DIFPLT, DPLRPL, GEOM, GEOMPC, RPLDPL
ESTOR	MAIN
FARP	MAIN, GEOM, GEOMC, GEOMPC, IMDIR, IMDIR, SOURCE, SOURCE
FEDDAT	SOURCE
FNANG	MAIN, GEOM, GEOMPC
FUDG	REFCYL, SCTCYL
FUDGI	RPLRCL, RPLSCL
FUDGJ	RCLRPL, SCLRPL
GEOMEL	MAIN, CAPINT, CYLINT, DFPTCL, DFRFT, DPLRCL, ENDIF, FKARG, FUNI, GEOMC, GEOMPC, NANG, RADCV, RCLDPL, RCLRPL, REFCAP, REFCYL, RFDFIN, RFDFPT, RFPTCL, RPLRCL, RPLSCL, SCLRPL, SCTCYL, TANG
GEOLPA	MAIN, DFPTW, DFRFT, DIFPLT, DPLRCL, DPLRPL, DPTNFW, GEOM, GEOMPC, IMAGE, IMDIR, PLAIN, RCLDPL, RCLRPL, REFBP, REFPLA, RFDFPT, RFPTCL, RPLDPL, RPLRCL, RPLRPL, SCLRPL
GROUND	MAIN, GEOM, PLAIN, RFPTCL
GTD	MAIN, FCT, RADCV, RPLSCL, SCLRPL, SCTCYL
HITPLT	DIFPLT, GEOM, PLAIN
IMAINF	MAIN, GEOM, GEOMPC, REFPLA, RFPTCL, RPLDPL, RPLRCL, RPLRPL, RPLSCL
IMCINF	GEOMC, GEOMPC, REFCAP
LDCBY	MAIN, GEOMPC
LOGDIF	MAIN, DIFPLT, DPLRPL, RPLDPL
LP.CY	MAIN, CYLINT, GEOMC, GEOMPC, PLAIN
LSHOP	GEOM, GEOMAC, PLAIN
LSHOT	MAIN, GEOM, GEOMPC
OUTPTO	MAIN, OUTPUT
PATDAT	MAIN, PATROT
PIS	MAIN, BLOCK DATA, BTANZ, CYLINT, DFPTCL, DFRFT, DI, DIFPLT, DPT, DPLRCL, DPLRPL, DZ, ENDIF, FCT, FFCT, FKARG, FKY, FRNELS, GEOM, GEOMC, GEOMPC, INCFLD, OUTPUT, PATROT, PFUN, PLAIN, PRIOUT, QFUN, RCLDPL, RCLRPL, REFBP, REFCAP, REFCYL, REFPLA, RFDFPT, RFPTCL, RPLDPL, RPLRCL, RPLRPL, RPLSCL, SCLRPL, SCTCYL, SOURCE, SOURCE, TANG
ROTROT	MAIN, ROTRAN
SORINF	MAIN, DFPTCL, DFRFT, DIFPLT, DPLRCL, DPLRPL, ENDIF, GEOM, GEOMC, GEOMPC, INCFLD, RCLDPL, RCLRPL, REFCAP, REFCYL, REFPLA, RFDFIN, RFDFPT, RFPTCL, RPLDPL, RPLRCL, RPLRPL, RPLSCL, SCLRPL, SCTCYL
SOURSF	MAIN, GEOM, GEOMC, GEOMPC, SOURCE, SOURCE
SRFACE	MAIN, GEOMC, GEOMPC
SURFAC	MAIN, DIFPLT, DPLRPL, GEOM, GEOMPC, RPLDPL
TEST	MAIN, CAPINT, CYLINT, DI, DIFPLT, DPT, DPLRCL, DPLRPL, ENDIF, FRNELS, GEOM, GEOMC, GEOMPC, PATROT, PLAIN, RCLDPL, RCLRPL, REFCAP, REFCYL, REFPLA, ROTRAN, RPLDPL, RPLRCL, RPLRPL, RPLSCL, SCLRPL, SCTCYL
THPHUV	MAIN, DIFPLT, DPLRCL, DPLRPL, ENDIF, INCFLD, RCLDPL, RCLRPL, REFCYL, RPLDPL, RPLRCL, SCLRPL
TOPD	BLOCK DATA, DI, DPT

C. Overlay Techniques

The Basic Scattering Code is a relatively large size computer code. Even though there are no big arrays, the large amount of coding requires a fairly significant amount of core in which to run. On an Amdahl computer, the code requires over 250 K bytes of core. The code has been designed, however, with overlay techniques in mind. Overlaying of the code can be accomplished by using the built in overlay techniques of a computer system, or by breaking the code up into smaller independent pieces that can be run separately with the results being superimposed later.

The code can be quite easily decomposed into three pieces. The three sections are composed of the subroutines necessary for plate fields, cylinder fields, and plate-cylinder interaction fields. The subroutines required to compute the plate fields are given in Table 4. If only plates are present in the geometry, the subroutines that are starred would not be necessary. The starred subroutines provide the shadowing algorithms for the cylinder. The subroutines required to compute the cylinder fields are given in Table 5. Similarly, the starred subroutines are necessary only if plate shadowing is desired. The subroutines required to compute the plate-cylinder interaction fields are given in Table 6. It is possible that for a particular problem or a particular computer system other techniques of separating the program would be more practical. The linkage information in Tables 2 and 3 should provide helpful information to accomplish such a task.

TABLE 4
SUBROUTINES USED IN PLATE COMPUTATIONS

BABS	IMAGE
BLOCK DATA	IMCDIR*
BLOG10	IMDIR
BTAN2	INCFLD
CAPINT*	OUTPUT
CYLINT*	PATROT
DEPTWD	PLAIN*
DI	PRIOUT
DIFPLT	REFBP
DPI	REFPLA
DPLRPL	ROTRAN
DW	RPLDPL
FECT	RPLRPL
FRNELS	SOURCE
GEOM	SOURCEP
GEOMC*	TANG*

*Non-essential unless a cylinder is present in the geometry.

TABLE 5
SUBROUTINES USED IN CYLINDER COMPUTATIONS

BABS	IMDIR*
BLOCK DATA	INCFLD
BLOG10	NANDB
BTAN2	OUTPUT
CAPINT	PATROT
CYLINT	PFUN
DEPTCL	PLAIN*
DQG32	POLYRT
DZ	PRIOUT
ENDIF	QFUN
FCT	RADCV
FKARG	REFCAP
FKY	REFCYL
FRNELS	RFPTCL
FUNI	ROTRAN
GEOM*	SCTCYL
GEOMC	SOURCE
IMAGE*	TANG
IMCDIR	

*Non-essential unless plates are present in the geometry.

TABLE 6
SUBROUTINES USED IN PLATE-CYLINDER INTERACTION COMPUTATIONS

BABS	NANDB
BLOCK DATA	OUTPUT
BLOG10	PATROT
BTAN2	PFUN
CAPINT	PLAINT
CYLINT	POLYRT
DFRFPT	PRIOUT
DI	QFUN
DPLRCL	RADCV
DPTNFW	RCLDPL
DQG32	RCLRPL
DW	REFBP
FCT	RDFIN
FKARG	RDFPT
FRNELS	RFPTCL
FUNI	ROTRAN
GEOM	RPLRCL
GEOMC	RPLSCL
GEOMPC	SCLRPL
IMAGE	SOURCE
IMCI IR	TANG
IMDIR	

D. Dimensions for Sources, Plates, and Edges

The maximum number of sources, plates, and edges that the Basic Scattering Code can accept is not limited by the theory. Any number of sources, plates, and edges can be used if the storage capacity of certain variables are set correctly in the DIMENSION statements and COMMON statements of the MAIN program and subroutines.

In order to change the maximum number of sources the code can accept, the dimension of the following variables in the MAIN program must be changed:

IMS(M_s), XSS(M_s ,3), VXSS(3,3, M_s), WM(M_s), WP(M_s), HS(M_s), HAWS(M_s),
THOZ(M_s), PHOZ(M_s), THOX(M_s), PHOX(M_s),

where M_s is equal to the maximum number of sources to be used. The variable MSDX should also be set equal to the integer M_s in the text of the code.

In order to change the number of plates and edges the code can accept, the dimensions of the following variables must be changed in the indicated subroutines' DIMENSION statements and in the COMMON statements. The location of all the commons can be found in Table 3. In MAIN, the dimensions of the variable XX(M_p , M_e ,3) should be changed, where M_p is equal to the maximum number of plates to be used and M_e is the maximum number of edges allowed on each plate. The value of the variable MPDX should be set equal to M_p and the variable MEDX should be set equal to M_e in the text. In subroutine GEOM, the dimension of the variable IHIT(M_e) should be changed. In subroutine RFPTCL the following dimensions should be changed:

IVD(M_{pp}), PHOR(M_{pp}), VRO(M_{pp}), PHORP(M_{pp})

where $M_{pp} = 2M_p + 1$.

In subroutine GEOMPC, the dimensions of the logical variable LCD(M_p , M_e) should be changed. In the subroutines RFDFPT and DFRFPT the dimensions of the following variables should be changed:

IVD(M_p , M_e), PHOR(M_p , M_e), THOR(M_p , M_e), VRO(M_p , M_e), URO(M_p , M_e),
PHORP(M_p , M_e).

The variables in the following common blocks should be changed:

BNDDCL: VDC(M_p, M_e), UDC(2), PDCR($M_p, M_e, 2$)
 TDCR($M_p, M_e, 2$), DTDC(M_p, M_e),
 BTDC($M_p, M_e, 4$), DDC($M_p, M_e, 2$)
 BNDFCL: BD($M_p, M_e, 2$)
 BNDICL: DTI(M_p), VTI($M_p, 2$), BTI($M_p, 4$)
 BNRCL: VCD(M_p, M_e), UCD(M_p, M_e), BCD($M_p, M_e, 2$)
 BRNPHW: PHWR(M_p, M_e)
 CLDRC: LDRC(M_p, M_e)
 CLRDC: LRDC(M_p, M_e)
 CLRFI: LRFI(M_p)
 CLRFS: LRFS(M_p)
 DOUBLE: IDD(361), ID(M_p, M_e), II
 EDMAG: VMAG(M_p, M_e)
 FNANG: FNP(M_p, M_e)
 GEOPLA: X($M_p, M_e, 3$), V($M_p, M_e, 3$),
 VP($m_p, M_e, 3$), VN($M_p, 3$),
 MEP(M_p), MPX
 IMAINF: XI($M_p, M_p, 3$), VXI(3, 3, M_p)
 LDCBY: LDC(M_p, M_e)
 LSHDP: LSTS, LSTD(M_p)
 LSHDT: LSHD(M_p), LIHD(M_p, M_p)
 SURFAC: LSURF(M_p)

E. Coordinate Systems

In order to simplify the variety of geometrical calculations performed in the code, a number of different coordinate systems are used. Each system allows a certain set of computations to be performed with maximum ease. Each of these systems are defined in terms of the reference coordinate system (RCS).

1. Reference coordinate system

The reference coordinate system (RCS) is the fundamental system of the code. The system geometry is defined and stored in the RCS. Many of the calculations carried out are done in the RCS. It is therefore also referred to as the "computational coordinate system". Each of the other coordinate systems is defined in terms of the RCS (using RCS coordinates or unit vectors).

This coordinate system is the fixed system that the user defines the input geometry with respect to. However, if a cylinder is defined using the RT: command, a new reference coordinate system is established, before the computations begin. The z-axis of the new system coincides with the cylinder axis, the x-axis with the "A" dimension of the cylinder, and the y-axis with the "B" dimension of the cylinder. All the other input geometries are rotated and/or translated to this new RCS or "cylinder coordinate system" using subroutine ROTRAN. This is done to simplify the computation of the cylinder and plate-cylinder interaction fields. This transformation is not visible to the user in terms of the input or output parameters. The term "reference coordinate system" is, therefore, used for both the original system or the new system without distinguishing between them.

2. Definition coordinate system

Normally, the system geometry is defined by the user in the reference coordinate system. However, the user may choose to perform a coordinate system transformation (using the RT: command) in order to define part of the geometry in some preferred coordinate system. In using the RT: command, the user creates a "definition coordinate system" to which the data which follows the command will pertain. The user defines the definition coordinate system by specifying the origin location and direction of the x_d and z_d axes unit vectors. The origin of the definition system is defined at point $TR = \hat{x} TR(1) + \hat{y} TR(2) + \hat{z} TR(3)$ in x,y,z RCS coordinates. The $\hat{x}_d$ unit vector of the definition system is defined by theta and phi angles THXP and PHXP as if it were a radial vector. The $\hat{z}_d$ unit vector is likewise defined by theta and phi angles THZP and PHZP in the RCS. The quantities TR, THXP, PHXP, THZP, and PHZP are all specified by the user. Note that all geometry defined in a definition coordinate system is immediately transformed into RCS notation using the method outlined in "transformation between systems" of this manual.

If the user defines the cylinder in a definition coordinate system (other than the RCS), the location of the system origin is stored along with the unit vectors of the definition system axes. The main program will later perform a transformation on the entire system geometry so that a new RCS is created where the z-axis of the new system coincides with cylinder axis. This new RCS is used for computational purposes.

3. Edge-fixed coordinate system

The code generates an edge-fixed coordinate system for each edge on every plate. The three (rectangular) coordinate system axes for each edge are positioned as follows:

1. in the plate plane and normal to the edge (the edge binormal)
2. normal to the plate
3. along the plate edge.

The unit vectors of the edge-fixed coordinate system axes are defined (using RCS unit vectors $\hat{x}$, $\hat{y}$, $\hat{z}$) as;

$$\hat{V}P = \text{edge binormal} = \hat{x} VP(MP,ME,1) + \hat{y} VP(MP,ME,2) + \hat{z} VP(MP,ME,3)$$

$$\hat{V}N = \text{plate normal} = \hat{x} VN(MP,1) + \hat{y} VN(MP,2) + \hat{z} VN(MP,3)$$

$$\hat{V} = \text{edge unit vector} = \hat{x} V(MP,ME,1) + \hat{y} V(MP,ME,2) + \hat{z} V(MP,ME,3).$$

Variables MP and ME specify which plate and which edge the unit vectors apply to.

The most significant use of the edge-fixed system is determining edge diffraction geometry. Incident and diffracted ray propagation angles along with polarization components are calculated by taking dot and cross products of edge-fixed unit vectors and ray propagation and polarization unit vectors. Edge-fixed unit vectors are also used in calculating geometry for intersecting plates, as well as checking to see if plates are flat.

The edge-fixed vectors are calculated in Section 2 of subroutine GEOM. Further details are given in this section.

4. Pattern cut coordinate system

The pattern cut coordinate system determines the axes about which the conical (theta fixed) or "orange-slice" (phi fixed) pattern cut is to be measured. It is also the coordinate system in which the code output is given.

The user defines the pattern cut coordinate system by specifying the theta and phi angles which define the x_p and z_p axes of the system in the RCS (THCX,PHCX,THCZ,PHCZ, respectively). The user then specifies the type of cut to be made (θ_p or ϕ_p cut) and the range and increment of angles (in PD: section of MAIN).

The pattern cut system axes are stored in a 3x3 matrix of components which define the axes unit vectors in RCS components (see "Transformation between systems"). This matrix is used in the MAIN program in converting specific pattern angles from pattern cut coordinate system notation to the reference coordinate system (subroutine PATROT). Note that the pattern cut coordinate system is subject to the mass geometry transformation that is performed in the MAIN program if the user defines a cylinder in a coordinate system other than the RCS (using the RT: command). This transformation, however is not visible to the user. Note also that definition of the pattern cut coordinate system is done independently of any RT: commands. The user always defines it in the RCS.

5. Reflection plate coordinate system

The reflection plate coordinate system is used to handle reflection from plates when image theory is not used (in subroutines DPLRPL, RCLRPL, and SCLRPL). Only two of the three rectangular axes unit vectors are used:

$\hat{V}_N$ = plate normal (calculated in subroutine GEOM)

$\hat{V}_T$ = vector tangent to plate and normal to incident (and reflected) ray propagation direction.

The unit vectors are used to convert polarization to and from reflection plate coordinate system (parallel and normal to plate).

6. Source coordinate system

The source coordinate system is the system in which the source is defined. There is one such system for each source, although only one appears in the computations at a given time. Each time another source is used the source coordinate system is redefined.

For a one-dimensional source, the dipole lies along the z_p axis. If an aperture source is used, the source lies in the x_p - z_p plane, centered about the origin. In both cases the source current flows in the $\hat{z}_p$ direction.

Note that in this code the source coordinate system is designated with the subscript "p". The system may also be referred to as the "primed" or "antenna" coordinate system.

The source coordinate system is defined by the user in the input part of the main program. It is redefined later in the main program within the source loop. The origin about which the source is centered, is located at $\bar{X}S = \hat{x} XS(1) + \hat{y} XS(2) + \hat{z} XS(3)$ in RCS components. The unit vectors of the system axes are defined by a 3x3 matrix $VXS(NI,NJ)$ of components. (See section on coordinate transformations). More specific definitions and illustrations are given in the section for subroutine SOURCE in the code manual.

7. Source image coordinate system

In many cases the code uses image theory in computing fields reflected from a plate. This involves computing an image source from which the reflected rays will appear to originate (see section on subroutine REFPLA). Assuming the source dimensions are known, the image source (or "source image") may be determined by computing the source image location (subroutine IMAGE) and the source image orientation (subroutine IMDIR). As the source location and orientation are specified using a source coordinate system, the "source image coordinate system" is used to define the image source. The location of the source image ($\bar{X}IS = \hat{x} XIS(1) + \hat{y} XIS(2) + \hat{z} XIS(3)$) is the origin of the source image coordinate system and the axes are defined by unit vectors in the same manner as the source coordinate system.

8. Transformation between systems

The majority of transformations performed in the code involve situations where it is necessary to transform a vector from one coordinate system to another. This involves rotation of coordinate systems without translation. Transformation of vector $\bar{V}$ in RCS components to $\bar{V}'$ in some other coordinate system is performed as follows:

$$\begin{bmatrix} V'_x \\ V'_y \\ V'_z \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

where $\bar{V} = \hat{x} V_x + \hat{y} V_y + \hat{z} V_z$ is the vector defined in the reference coordinate system and $\bar{V}' = \hat{x}' V'_x + \hat{y}' V'_y + \hat{z}' V'_z$ is the vector defined in the second coordinate system. Inverse transformations are done by multiplying the vector $\bar{V}'$ by the transpose of the rotation matrix as follows:

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} x_{11} & x_{21} & x_{31} \\ x_{12} & x_{22} & x_{32} \\ x_{13} & x_{23} & x_{33} \end{bmatrix} \begin{bmatrix} v'_x \\ v'_y \\ v'_z \end{bmatrix} .$$

The unit vectors of the second coordinate system are defined in the reference coordinate system as follows:

$$\begin{aligned} \hat{x}' &= \hat{x} x_{11} + \hat{y} x_{12} + \hat{z} x_{13} \\ \hat{y}' &= \hat{x} x_{21} + \hat{y} x_{22} + \hat{z} x_{23} \\ \hat{z}' &= \hat{x} x_{31} + \hat{y} x_{32} + \hat{z} x_{33} . \end{aligned}$$

The matrix used to transform (rotate) the vector is generally referred to as "x,y,z components defining the (name of system) coordinate system axes unit vectors in RCS components". The individual matrix elements are determined as follows:

$$x_{mn} = \hat{x}'_m \cdot \hat{x}_n \quad (\text{i.e., } x_{13} = \hat{x}'_1 \cdot \hat{z}, \text{ etc}).$$

When transforming a point, it is necessary to perform a translation as well as a rotation of coordinate systems (if the origins of the systems are different). To handle these situations, the code translates the point before rotating. Transformation of point $\vec{P} = \hat{x} P_x + \hat{y} P_y + \hat{z} P_z$ in the reference coordinate system to point $\vec{P}' = \hat{x}' P'_x + \hat{y}' P'_y + \hat{z}' P'_z$ in another coordinate system is done as follows:

$$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} P_x - x_{0x} \\ P_y - x_{0y} \\ P_z - x_{0z} \end{bmatrix} ,$$

where $\vec{x}_0 = \hat{x} x_{0x} + \hat{y} x_{0y} + \hat{z} x_{0z}$ is the location of the origin of the second coordinate system in the reference coordinate system and $x_{11}, x_{12}, x_{13}, \text{ etc.}$ are as defined previously. The reverse transformation is performed as follows:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} x_{0x} \\ x_{0y} \\ x_{0z} \end{bmatrix} + \begin{bmatrix} x_{11} & x_{21} & x_{31} \\ x_{12} & x_{22} & x_{32} \\ x_{13} & x_{23} & x_{33} \end{bmatrix} \begin{bmatrix} P'_x \\ P'_y \\ P'_z \end{bmatrix} .$$

CHAPTER IV CODE DESCRIPTION

This section is divided into two sections, the first of which describes the operation of the MAIN program in detail. The second describes the subroutine and function operations. For each subroutine the following is given (as appropriate):

1. a statement of purpose
2. an illustration showing the geometry
3. a brief narrative of the method used
4. a flow diagram
5. a dictionary of major variables
6. a listing of the code.

The comment statements in the code listings follow the statements of the flow diagrams, simplifying correlation of the two.

MAIN

PURPOSE

The main program reads the system geometry given by the user and directs the calculation of the scattered fields.

The main program is broken down into three parts as follows:

1. Data input section
2. Input conversion section
3. Main computation section.

Each of these sections is outlined on the following pages.

1. Data Input Section

The data input section reads an input file that contains the data specifying the geometry of the problem to be considered and prepares it for use by the code. The data input section is described in the User's Manual [8]. The commands available and the general flow of the input section is given in detail there, and will therefore not be repeated here.

```

1 C!!!
2 C!!! THIS PROGRAM WAS WRITTEN AT THE OHIO STATE UNIVERSITY
3 C!!! ELECTROSCIENCE LABORATORY. ANY PROBLEMS OR COMMENTS
4 C!!! CAN BE REFERRED TO:
5 C!!!
6 C!!! WALTER D. BURNSIDE (OR) RONALD J. MARHEFKA
7 C!!! ELECTROSCIENCE LABORATORY
8 C!!! 1320 KIMMEAR RD.
9 C!!! COLUMBUS, OHIO 43212
10 C!!! PHONE: (614) 422-5747 (OR) 422-5752
11 C!!!
12 C!!! THIS PROGRAM COMPUTES THE FAR FIELD PATTERN OF AN ANTENNA
13 C!!! OR SET OF ANTENNAS IN THE PRESENCE OF A SET OF PLATES
14 C!!! AND/OR IN THE PRESENCE OF AN ELLIPTIC CYLINDER.
15 C!!! THE PLATES ARE DEFINED BY THEIR CORNER LOCATIONS.
16 C!!! AS DIMENSIONED, IT CAN HANDLE 14 PLATES WITH A MAXIMUM
17 C!!! OF 8 CORNERS PER PLATE, AND 50 ANTENNA ELEMENTS CAN
18 C!!! BE INPUT. THE CYLINDER IS DEFINED BY ITS RADIUS
19 C!!! ON ITS MAJOR AND MINOR AXIS AND THE END CAPS ARE
20 C!!! DEFINED BY THEIR POSITION ON THE CYLINDER AXIS AND
21 C!!! THE ANGLE OF THEIR SURFACES WITH THE CYLINDER AXIS IN
22 C!!! THE X-Z CYL. PLANE. NOTE THAT THE LIMITS ON THE NUMBER
23 C!!! OF PLATES, CORNERS, AND SOURCES ARE ONLY DUE TO THE SIZE
24 C!!! OF THE ARRAYS. THE LINEAR DIMENSIONS ARE INPUT IN METERS
25 C!!! UNLESS SPECIFIED. THE ANGULAR DIMENSIONS ARE IN DEGREES.
26 C!!!
27 C!!! NOTE THAT COMMENTS ARE GIVEN IN TWO FORMS:
28 C!!! C!!! IMPLIES EXPLANATION OF PROGRAM SECTION
29 C!!! IN TERMS OF THEORY
30 C!!! C$$$ IMPLIES DESCRIPTION OF INPUT DATA
31 C!!! C--- IMPLIES COMMAND INPUT READ SECTION
32 C!!!
33 C!!! THIS VERSION WAS WRITTEN 8/2/79
34 C!!!
35 C!!!
36 COMPLEX EITH,EIPH,ERTH,ERPH,ERPCT,ERPCP,ESTH,ESPH
37 COMPLEX ERPST,ERPSP,ERTH,ERPPH,ERRPT,ERRPP,ERCPT,ERCPP
38 COMPLEX CJ,CPI4,WI,EIH,EPH,ETHT(361),EPHT(361)
39 COMPLEX EDCTH,EDCPH,EDPTH,EDPPH,ERPD,ERPD,EDCRPT,EDCRPP
40 COMPLEX EDWPT,EDWPP,ERDTH,ERDPH,EDRCT,EDRCP,ERCAT,ERCAP
41 COMPLEX EDPCTH,EDPCPH,ERPDCT,ERPDCP,EDDTH,EDDPH,ERSPT,ERSPP
42 DIMENSION INS(50),XSS(50,3),VXSS(3,3,50),WV(50),WP(50)
43 DIMENSION HS(50),HWS(50),THOZ(50),PHOZ(50),THOX(50),PHOX(50)
44 DIMENSION XX(14,6,3),XCO(3),XXCO(3),XOO(3),XXX(3)
45 DIMENSION XPD(3),YPD(3),ZPD(3)
46 DIMENSION JMN(3),JMX(3)
47 DIMENSION LABEL(2,3),UNIT(3),IR(24),IT(14),ITT(10)
48 DIMENSION XOR(3),TR(3),XP(3),YP(3),ZP(3),XO(3),XPP(3)
49 LOGICAL LSOP,LOUT,LSRFC,LSURF,LSHD,LCYL,LPLA,LNROT
50 LOGICAL LIHD,LDEBUB,LTEST,LSLOPE,LCORNR,LDC
51 LOGICAL LWRITE,LPLT,LGRND,LAMP,LPRAD,LRANG,LCNPAT
52 LOGICAL LRFC,LRFI,LRFS,LRDC,LRDC
53 COMMON/DOUBLE/ID(361),ID(14,6),II
54 COMMON/FARP/IN,H,HAW
55 COMMON/SOURCE/FACTOR
56 COMMON/TEST/LDEBUB,LTEST
57 COMMON/LOGDIF/LSLOPE,LCORNR
58 COMMON/SORINF/XS(3),VXS(3,3)
59 COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
60 COMMON/PIS/PI,TPI,DPR,RPD
61 COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
62 COMMON/COMP/CJ,CPI4
63 COMMON/THPHUV/DT(3),DP(2)
64 COMMON/SURFAC/LSURF(14)

```

```

65      COMMON/LSRACC/LSRACC(2)
66      COMMON/LSHDT/LSHDT(14),LIHD(14,14)
67      COMMON/LDCBY/LDC(14,6)
68      COMMON/FNANG/FNP(14,6)
69      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
70      2,MEPC(14),MPX
71      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
72      COMMON/CTD/AS,IGC,SAS,SASP,CAS
73      COMMON/ESTOR/ETHT,EPHT
74      COMMON/LPLCY/LPLA,LCYL
75      COMMON/GROUND/LGRND,MPXR
76      COMMON/OUTPTD/LPRAD,LRANG,PRAD,RANG,WL
77      COMMON/PATDAT/XPC(3),YPC(3),ZPC(3)
78      COMMON/ROTROT/XCL(3),YCL(3),ZCL(3)
79      COMMON/CLRFC/LRFC
80      COMMON/CLRFI/LRFI(14)
81      COMMON/CLRFS/LRFS(14)
82      COMMON/CLDRC/LDRC(14,6)
83      COMMON/CLRDC/LRDC(14,6)
84      DATA UNIT/1.,.3048,.00254/
85      DATA LABEL/'MET','ERS','FEE','T','INC','HES'/
86      DATA IT/'TO','PD','PG','SG','LP','PP','GP','XO','RT'/
87      2,'CG','AM','RG','CM','CE'/
88      DATA IIT/'UN','FR','NX','EI','NP','NC','NG','NO','PR'/
89      2,'US'/
90 C!!! MAX. DIMENSION OF SOURCES, PLATES, AND EDGES.
91      MSDX=50
92      MPDX=14
93      MEDX=0
94 C!!! NOTE: IN SUB. REPTCL THE VARIABLES IVD,PHOR,PHORP,AND VRO
95 C!!! MUST BE DIMENSIONED 2*MPDX+1
96      GO TO 2701
97 2700 CONTINUE
98      WRITE(6,3006)
99      WRITE(6,3005)
100 2701 CONTINUE
101 C!!! INITIALIZE DATA TO DEFAULT VALUES.
102      LDEBUG=.FALSE.
103      LTEST=.FALSE.
104      LOUT=.FALSE.
105      LSLOPE=.TRUE.
106      LCONNR=.TRUE.
107      LSOR=.FALSE.
108      LCYL=.FALSE.
109      LPLA=.FALSE.
110      LGRND=.FALSE.
111      LWRITE=.TRUE.
112      LPLT=.FALSE.
113      LAMP=.FALSE.
114      LPRAD=.FALSE.
115      LRANG=.FALSE.
116      RANG=1.
117      PRAD=0.
118      JMX(1)=1
119      JMX(1)=7
120      JMX(2)=1
121      JMX(2)=3
122      JMX(3)=1
123      JMX(3)=4
124      LCONPAT=.TRUE.
125      TPPD=90.
126      THCZ=0.
127      PHCZ=0.
128      THCX=90.
129      PHCX=0.
130      XPD(1)=1.

```

```

131 XPD(2)=0.
132 XPD(3)=0.
133 YPD(1)=0.
134 YPD(2)=1.
135 YPD(3)=0.
136 ZPD(1)=0.
137 ZPD(2)=0.
138 ZPD(3)=1.
139 IB=0
140 IE=300
141 IS=1
142 FREQ=.2597925
143 MPX=0
144 MEP(1)=4
145 XX(1,1,1)=1.
146 XX(1,1,2)=1.
147 XX(1,1,3)=0.
148 XX(1,2,1)=-1.
149 XX(1,2,2)=1.
150 XX(1,2,3)=0.
151 XX(1,3,1)=-1.
152 XX(1,3,2)=-1.
153 XX(1,3,3)=0.
154 XX(1,4,1)=1.
155 XX(1,4,2)=-1.
156 XX(1,4,3)=0.
157 MSX=0
158 XSS(1,1)=0.
159 XSS(1,2)=0.
160 XSS(1,3)=1.
161 IMS(1)=0
162 HS(1)=0.5
163 HAWS(1)=0.
164 THOZ(1)=0.
165 PHOZ(1)=0.
166 THOX(1)=0.
167 PHOX(1)=0.
168 VXSS(1,1,1)=1.
169 VXSS(1,2,1)=0.
170 VXSS(1,3,1)=0.
171 VXSS(2,1,1)=0.
172 VXSS(2,2,1)=1.
173 VXSS(2,3,1)=0.
174 VXSS(3,1,1)=0.
175 VXSS(3,2,1)=0.
176 VXSS(3,3,1)=1.
177 WM(1)=1.
178 WP(1)=0.
179 RADIUS=3.
180 IPLT=3
181 THZP=0.
182 PHZP=0.
183 THXP=0.
184 PHXP=0.
185 TR(1)=0.
186 TR(2)=0.
187 TR(3)=0.
188 XP(1)=1.
189 XP(2)=0.
190 XP(3)=0.
191 YP(1)=0.
192 YP(2)=1.
193 YP(3)=0.
194 ZP(1)=0.
195 ZP(2)=0.
196 ZP(3)=1.

```

```

197      AA=1.
198      BB=1.
199      ZCN=-3.
200      THTN=90.
201      ZCP=3.
202      THTP=90.
203      XXCO(1)=0.
204      XXCO(2)=0.
205      XXCO(3)=0.
206      XCL(1)=1.
207      XCL(2)=0.
208      XCL(3)=0.
209      YCL(1)=0.
210      YCL(2)=1.
211      YCL(3)=0.
212      ZCL(1)=0.
213      ZCL(2)=0.
214      ZCL(3)=1.
215      IUNIT=1
216      UNITS=UNIT(IUNIT)
217      IUNST=0
218      IUNSP=IUNST
219      GO TO 2999
220 3000 CONTINUE
221      WRITE(6,3006)
222 3006 FORMAT(1X,1H*,76X,1H*)
223      WRITE(6,3006)
224      WRITE(6,3005)
225 3005 FORMAT(1X,26(3H***))
226 C!!! READ IN VARIOUS COMMAND OPTIONS.
227 2999 READ(5,3001,END=3004)(IR(I),I=1,24)
228 3001 FORMAT(24A3)
229      WRITE(6,3002)
230 3002 FORMAT(1H ,////////,1X,26(3H***))
231      WRITE(6,3006)
232      WRITE(6,3003)(IR(I),I=1,24)
233 3003 FORMAT(1X,1H*,2X,24A3,2X,1H*)
234      IF(IR(1).EQ.IT(13))GO TO 3090
235      IF(IR(1).EQ.IT(14))GO TO 3000
236      WRITE(6,3006)
237      WRITE(6,3006)
238 C!!!
239 C!!! CHECK AGAINST STORED OPTIONS
240 C!!!
241 C!!! CM: COMMENT CARD
242 C!!! CE: LAST COMMENT CARD
243 C!!! TO: TEST DATA GENERATION OPTION.
244 C!!! UN: UNITS OF INPUT
245 C!!! US: UNITS OF HS AND HAWS IN SG:
246 C!!! FR: FREQUENCY
247 C!!! PD: PATTERN DATA DESIRED
248 C!!! PG: PLATE GEOMETRY INPUT
249 C!!! SG: SOURCE GEOMETRY INPUT
250 C!!! AM: NEC OR AMP INPUT
251 C!!! PR: POWER RADIATED INPUT
252 C!!! LP: LINE PRINTER LISTING OF RESULTS
253 C!!! PP: PEN PLOT OF RESULTS
254 C!!! GP: INCLUDE INFINITE GROUND PLANE
255 C!!! XO: EXECUTE PROGRAM
256 C!!! RT: TRANSLATE AND/OR ROTATE COORDINATES
257 C!!! CG: CYLINDER GEOMETRY INPUT
258 C!!! RG: FAR FIELD RANGE INPUT
259 C!!! NP: NEXT SET OF PLATES
260 C!!! NG: NO GROUND PLANE
261 C!!! NC: NO CYLINDER
262 C!!! NS: NEXT SET OF SOURCES

```

```

263 C!!! NX: NEXT PPOPLEM
264 C!!! EN: END PROGRAM
265 C!!!
266 IF(IR(1).EQ.IT(1))GO TO 3100
267 IF(IR(1).EQ.IT(2))GO TO 3200
268 IF(IR(1).EQ.IT(12)) GO TO 3250
269 IF(IR(1).EQ.IT(3))GO TO 3300
270 IF(IR(1).EQ.IT(4))GO TO 3400
271 IF(IR(1).EQ.IT(11))GO TO 3450
272 IF(IR(1).EQ.IT(5))GO TO 3500
273 IF(IR(1).EQ.IT(6))GO TO 3600
274 IF(IR(1).EQ.IT(7))GO TO 3700
275 IF(IR(1).EQ.IT(8))GO TO 3800
276 IF(IR(1).EQ.IT(9))GO TO 3900
277 IF(IR(1).EQ.IT(10))GO TO 4000
278 IF(IR(1).EQ.ITT(1)) GO TO 4100
279 IF(IR(1).EQ.ITT(2)) GO TO 4200
280 IF(IR(1).EQ.ITT(3)) GO TO 2700
281 IF(IR(1).EQ.ITT(4)) GO TO 997
282 IF(IR(1).EQ.ITT(5)) GO TO 3350
283 IF(IR(1).EQ.ITT(6)) GO TO 4050
284 IF(IR(1).EQ.ITT(7)) GO TO 3750
285 IF(IR(1).EQ.ITT(8)) GO TO 3490
286 IF(IR(1).EQ.ITT(9)) GO TO 3440
287 IF(IR(1).EQ.ITT(10)) GO TO 4110
288 WRITE(6,3021)
289 3021 FORMAT(' *** PROGRAM ABORTS!!! COMMAND INPUT IS NOT PART',
290 1' OF STORED COMMAND LIST ***')
291 3004 STOP
292 C-----
293 1090 CONTINUE
294 C----- CM: CE: COMMANDS -----
295 C$$$
296 C$$$ IR(1)=CM: OR CE: FOLLOWED BY AN ALPHANUMERIC STRING OF
297 C$$$ CHARACTERS. THE CM: COMMAND IMPLIES THAT THERE WILL BE
298 C$$$ ANOTHER COMMENT CARD FOLLOWING IT. THE LAST COMMENT CARD
299 C$$$ MUST HAVE THE CE: COMMAND ON IT. IF THERE IS ONLY ONE
300 C$$$ COMMENT CARD THE CE: COMMAND SHOULD BE USED.
301 C$$$
302 READ(5,3001) (IR(I),I=1,24)
303 WRITE(6,3003) (IR(I),I=1,24)
304 IF(IR(1).EQ.IT(14)) GO TO 3000
305 IF(IR(1).EQ.IT(13)) GO TO 3090
306 WRITE(6,3091)
307 3091 FORMAT(' *** PROGRAM ABORTS!!! CE: COMMAND MUST BE',
308 2' USED TO END COMMENTS. ***')
309 STOP
310 C-----
311 1100 CONTINUE
312 C----- TO: COMMAND -----
313 C$$$
314 C$$$ LDEBUG=DEBUG DATA OUTPUT ON LINE PRINTER(TRUE OR FALSE)
315 C$$$
316 C$$$ LTEST=TEST DATA TO INSURE PROGRAM OPERATION(TRUE OR FALSE)
317 C$$$
318 C$$$ LOUT=OUTPUT MAIN PROGRAM DATA ON LINE PRINTER(TRUE OR FALSE)
319 C$$$
320 READ(5,*) LDEBUG,LTEST,LOUT
321 WRITE(6,3101)LDEBUG,LTEST,LOUT
322 1101 FORMAT(2H *,5X,'LDEBUG= ',L3,5X,'LTEST= ',L3,5X,'LOUT= ',L3,
323 1179,1H*)
324 WRITE(6,3006)
325 C$$$
326 C$$$ LSLOPE=SLOPE DIFFRACTED FIELD DESIRED (T OR F)
327 C$$$
328 C$$$ LCONNR=CORNER DIFFRACTED FIELD DESIRED (T OR F)

```

```

329 C$$$
330 C$$$ LSOR=ANTENNA PATTERN ALONE(TRUE OR FALSE)
331 C$$$
332 READ(5,*)LSLOPE,LCORNR,LSOR
333 WRITE(6,3102)LSLOPE,LCORNR,LSOR
334 3102 FORMAT(2H *,5X,'LSLOPE= ',L3,5X,'LCORNR= ',L3,5X,'LSOR= ',L3,
335 1T79,1H*)
336 WRITE(6,3006)
337 IF(LSOR)WRITE(6,3402)
338 3402 FORMAT(2H *,5X,'SOURCE PATTERN ALONE IS COMPUTED!!!',T79,1H*)
339 IF(LSOR)WRITE(6,3006)
340 C$$$
341 C$$$ JMN(1),JMX(1)=OPTION TO VARIOUS RAY TERMS FOR PLATES:
342 C$$$ 0=SKIP PLATES SECTION
343 C$$$ 1=INCIDENT FIELD
344 C$$$ 2=SINGLE REFLECTED FIELD
345 C$$$ 3=DOUBLE REFLECTED FIELD
346 C$$$ 4=SINGLE DIFFRACTED FIELD
347 C$$$ 5=REFLECTED/DIFFRACTED FIELD
348 C$$$ 6=DIFFRACTED/REFLECTED FIELD
349 C$$$ NOTE: NORMALLY JMN(1)=1 AND JMX(1)=7. THIS COMPUTES ALL FIELD
350 C$$$ VALUES INCLUDING IDENTIFYING DOUBLE DIFFRACTION PROBLEM AREAS
351 C$$$ FOR A CONVEX OR CONCAVE PLATE STRUCTURE.
352 C$$$
353 C$$$ JMN(2),JMX(2)=OPTION TO RUN VARIOUS RAY TERMS FOR CYLINDER:
354 C$$$ 0=SKIP CYLINDER SECTION
355 C$$$ 1=INCIDENT,REFLECTED,TRANSITION,AND CREEPING WAVE FIELDS
356 C$$$ 2=SINGLE REFLECTED FIELDS FROM END CAPS
357 C$$$ 3=SINGLE DIFFRACTED FIELDS FROM END CAP RIMS
358 C$$$ NOTE: NORMALLY JMN(2)=1 AND JMX(2)=3. THIS COMPUTES ALL FIELD
359 C$$$ VALUES FOR A FINITE ELLIPTIC CYLINDER.
360 C$$$
361 C$$$ JMN(3),JMX(3)=OPTION TO RUN VARIOUS RAY TERMS FOR
362 C$$$ PLATE-CYLINDER INTERACTIONS:
363 C$$$ 0=SKIP PLATE-CYLINDER INTERACTION SECTION
364 C$$$ 1=FIELDS REFLECTED FROM THE PLATES THEN REFLECTED OR
365 C$$$ DIFFRACTED FROM THE CYLINDER
366 C$$$ 2=FIELDS REFLECTED OR DIFFRACTED FROM THE CYLINDER THEN
367 C$$$ REFLECTED FROM THE PLATES
368 C$$$ 3=FIELDS REFLECTED FROM THE CYLINDER THEN DIFFRACTED
369 C$$$ FROM THE PLATES
370 C$$$ 4=FIELDS DIFFRACTED FROM THE PLATES THEN REFLECTED
371 C$$$ FROM THE CYLINDER
372 C$$$ NOTE: NORMALLY JMN(3)=1 AND JMX(3)=4.
373 C$$$
374 READ(5,*) JMN(1),JMX(1),JMN(2),JMX(2),JMN(3),JMX(3)
375 IF(JMN(1).LT.0) JMN(1)=1
376 IF(JMX(1).GT.7) JMX(1)=7
377 IF(JMN(2).LT.0) JMN(2)=1
378 IF(JMX(2).GT.3) JMX(2)=3
379 IF(JMN(3).LT.0) JMN(3)=1
380 IF(JMX(3).GT.4) JMX(3)=4
381 IF(LSOR) JMN(1)=1
382 IF(LSOR) JMX(1)=1
383 WRITE(6,3103) JMN(1),JMX(1),JMN(2),JMX(2),JMN(3),JMX(3)
384 3103 FORMAT(2H *,2X,'JMN(1)= ',I2,2X,'JMX(1)= ',I2,2X,
385 2,'JMN(2)= ',I2,2X,'JMX(2)= ',I2,2X,'JMN(3)= ',I2,2X
386 2,'JMX(3)= ',I2,T79,1H*)
387 GO TO 3000
388 C-----
389 4100 CONTINUE
390 C--- UN: COMMAND -----
391 C$$$
392 C$$$ IUNIT=INDICATOR OF UNITS USED FOR INPUT DATA.
393 C$$$ 1=METERS
394 C$$$ 2=FEET

```

```

395 C$$$      3=INCHES
396 C$$$
397      READ(5,*) IUNIT
398      UNITS=UNIT(IUNIT)
399      WRITE(6,4101) (LABEL(J,IUNIT),J=1,2)
400 4101      FORMAT(2H *,5X,'ALL THE LINEAR DIMENSIONS BELOW ARE'
401          2,' ASSUMED TO BE IN ',2A3,T79,1H*)
402          GO TO 3600
403 C-----
404 4110      CONTINUE
405 C---      US:  COMMAND  -----
406 C$$$
407 C$$$      IUNST=INDICATOR OF UNITS USED FOR HS AND HAWS IN THE
408 C$$$      SG:  COMMAND.
409 C$$$          0=WAVELENGTHS
410 C$$$          1=METERS
411 C$$$          2=FEET
412 C$$$          3=INCHES
413 C$$$
414 C$$$      NOTE: IF ONE SOURCE IS SPECIFIED IN WAVELENGTHS, THEY ALL
415 C$$$          MUST BE IN WAVELENGTHS.
416      READ(5,*) IUNST
417      IF(MSX.EQ.0) GO TO 4112
418      IF(IUNST.EQ.0.AND.IUNSP.EQ.0) GO TO 4112
419      IF(IUNST.NE.0.AND.IUNSP.NE.0) GO TO 4112
420      WRITE(6,4111)
421 4111      FORMAT(' *** PROGRAM ABORTS IN SOURCE UNITS. ALL UNITS NOT'
422          2,' SPECIFIED IN WAVELENGTHS!!! ***')
423          STOP
424 4112      CONTINUE
425      IF(IUNST.EQ.0) GO TO 4114
426      WRITE(6,4113) (LABEL(J,IUNST),J=1,2)
427 4113      FORMAT(2H *,5X,'THE SOURCE LENGTH HS AND WIDTH HAWS ARE'
428          2,' ASSUMED TO BE IN ',2A3,T79,1H*)
429          GO TO 4116
430 4114      WRITE(6,4115)
431 4115      FORMAT(2H *,5X,'THE SOURCE LENGTH HS AND WIDTH HAWS ARE'
432          2,' ASSUMED TO BE IN WAVELENGTHS',T79,1H*)
433 4116      IUNSP=IUNST
434          GO TO 3600
435 C-----
436 4200      CONTINUE
437 C---      FR:  COMMAND  -----
438 C$$$
439 C$$$      FREQ=FREQUENCY IN GIGAHERTZ.
440 C$$$
441      READ(5,*) FREQ
442      WL=.2997925/FREQ
443      WRITE(6,4201) FREQ
444 4201      FORMAT(2H *,5X,'FREQUENCY= ',F7.3,' GIGAHERTZ',T79,1H*)
445      WRITE(6,3006)
446      WRITE(6,4202) WL
447 4202      FORMAT(2H *,5X,'WAVELENGTH= ',F10.6,' METERS',T79,1H*)
448          GO TO 3600
449 C-----
450 3200      CONTINUE
451 C---      PD:  COMMAND  -----
452 C$$$
453 C$$$      THCZ,PHCZ=ORIENTATION OF THE ZPD AXIS RELATIVE TO THE
454 C$$$      FIXED COORDINATE SYSTEM.
455 C$$$
456 C$$$      THCX,PHCX=ORIENTATION OF THE XPD AXIS RELATIVE TO THE
457 C$$$      FIXED COORDINATE SYSTEM
458 C$$$
459      READ(5,*) THCZ,PHCZ,THCX,PHCX
460      ZPD(1)=SIN(THCZ*RPD)*COS(PHCZ*RPD)

```

```

461 ZPD(2)=SIN(THCZ*RPD)*SIN(PHCZ*RPD)
462 ZPD(3)=COS(THCZ*RPD)
463 XPD(1)=SIN(THCX*RPD)*COS(PHCX*RPD)
464 XPD(2)=SIN(THCX*RPD)*SIN(PHCX*RPD)
465 XPD(3)=COS(THCX*RPD)
466 C!!! INSURE XPD IS PERPENDICULAR TO ZPD
467 DZX=ZPD(1)*XPD(1)+ZPD(2)*XPD(2)+ZPD(3)*XPD(3)
468 IF(ABS(DZX).GT.0.1) WRITE(6,3201)
469 3201 FORMAT(' *** PROGRAM ABORTS IN PATTERN CUT SECTION.')
470 2,' THE COORDINATES ARE NOT ORTHOGONAL!!! ***')
471 IF(ABS(DZX).GT.0.1) STOP
472 XPD(1)=XPD(1)-ZPD(1)*DZX
473 XPD(2)=XPD(2)-ZPD(2)*DZX
474 XPD(3)=XPD(3)-ZPD(3)*DZX
475 DOT=XPD(1)*XPD(1)+XPD(2)*XPD(2)+XPD(3)*XPD(3)
476 DOT=SQRT(DOT)
477 XPD(1)=XPD(1)/DOT
478 XPD(2)=XPD(2)/DOT
479 XPD(3)=XPD(3)/DOT
480 YPD(1)=ZPD(2)*XPD(3)-ZPD(3)*XPD(2)
481 YPD(2)=ZPD(3)*XPD(1)-ZPD(1)*XPD(3)
482 YPD(3)=ZPD(1)*XPD(2)-ZPD(2)*XPD(1)
483 WRITE(6,3202)
484 3202 FORMAT(2H *,5X,'THE PATTERN AXES ARE AS FOLLOWS:',T79,1H*)
485 WRITE(6,3006)
486 WRITE(6,3203) (XPD(N),N=1,3)
487 3203 FORMAT(2H *,5X,'XPD(1)=',F10.5,' XPD(2)=',F10.5,' XPD(3)=',
488 2,F10.5,T79,1H*)
489 WRITE(6,3006)
490 WRITE(6,3204) (YPD(N),N=1,3)
491 3204 FORMAT(2H *,5X,'YPD(1)=',F10.5,' YPD(2)=',F10.5,' YPD(3)=',
492 2,F10.5,T79,1H*)
493 WRITE(6,3006)
494 WRITE(6,3205) (ZPD(N),N=1,3)
495 3205 FORMAT(2H *,5X,'ZPD(1)=',F10.5,' ZPD(2)=',F10.5,' ZPD(3)=',
496 2,F10.5,T79,1H*)
497 C$$$
498 C$$$ LCNPAT=IS PATTERN CONIC CUT(T OR F)?
499 C$$$ T=THETA CUT(CONIC CUT)
500 C$$$ F=PHI CUT(PHI CONSTANT)
501 C$$$
502 C$$$ TPPD=PATTERN ANGLE THAT IS CONSTANT
503 C$$$ IF LCNPAT=T: TPPD=THP CONSTANT
504 C$$$ IF LCNPAT=F: TPPD=PHP CONSTANT
505 C$$$
506 READ(5,*) LCNPAT,TPPD
507 WRITE(6,3006)
508 IF(.NOT.LCNPAT) WRITE(6,3206) TPPD
509 3206 FORMAT(2H *,5X,'THETA IS BEING VARIED WITH PHI=',F10.5,
510 2,T79,1H*)
511 IF(LCNPAT) WRITE(6,3207) TPPD
512 3207 FORMAT(2H *,5X,'PHI IS BEING VARIED WITH THETA=',F10.5,
513 2,T79,1H*)
514 WRITE(6,3006)
515 C$$$
516 C$$$ IB,IE,IS=BEGIN,END,STEP
517 C$$$
518 READ(5,*) IB,IE,IS
519 IF(IE.LT.0) IB=0
520 IF(IE.GT.360) IE=360
521 IF(IS.LE.0) IS=1
522 WRITE(6,3208) IB,IE,IS
523 3208 FORMAT(2H *,5X,'THE RANGE OF PATTERN ANGLE INDICES FOR THIS'
524 2,' RUN ARE: ',I3,2(' ',I3),T79,1H*)
525 GO TO 3000
526 C-----

```

```

527 3250 CONTINUE
528 C--- RG* COMMAND -----
529 C$$$
530 C$$$ RANGS=FAR FIELD RANGE DISTANCE
531 C$$$
532 C$$$ NOTE IF RANGS IS GREATER THAN OR EQUAL TO 1.E30
533 C$$$ THAN LRANG WILL BE SET FALSE
534 C$$$
535 LRANG=.TRUE.
536 READ(5,*) RANGS
537 IF(RANGS.GT.9.9E29) GO TO 3252
538 RANG=UNITS*RANGS
539 WRITE(6,3251) RANGS,(LABEL(J,IUNIT),J=1,2),RANG
540 3251 FORMAT(2H *,5X,'THE FAR FIELD RANGE SPECIFIED IS ',E12.6,
541 2' IN ',2A3,T79,1H*,/2H *,5X,'THE RANGE SPECIFIED IN METERS'
542 2,' IS ',E12.6,T79,1H*)
543 GO TO 3250
544 3252 CONTINUE
545 LRANG=.FALSE.
546 RANG=1.
547 WRITE(6,3253)
548 3253 FORMAT(2H *,5X,'NO FAR FIELD RANGE SPECIFIED.',T79,1H*)
549 GO TO 3250
550 C-----
551 3300 CONTINUE
552 C--- PG* COMMAND -----
553 C$$$
554 C$$$ PLATE GEOMETRY INPUT
555 C$$$
556 LPLA=.TRUE.
557 MPX=MPX+1
558 IF (MPX.GT.MPDX) WRITE(6,901) MPX
559 901 FORMAT (' ***** NUMBER OF PLATES= ',I3,' PROGRAM ABORTS',
560 2' SINCE MAX. PLATE DIMENSION IS EXCEEDED. *****')
561 IF (MPX.GT.MPDX) STOP
562 WRITE(6,3301)MPX
563 3301 FORMAT(2H *,5X,'THIS IS PLATE NO. ',I3,' IN THIS ',
564 1'SIMULATION.',T79,1H*)
565 MP=MPX
566 WRITE(6,3000)
567 WRITE(6,3000)
568 WRITE(6,3000)
569 C$$$
570 C$$$ MEP(MP)=NUMBER OF CORNERS ON THE MP-TH PLATE.
571 C$$$
572 READ(5,*) MEP(MP)
573 MEX=MEP(MP)
574 IF (MEX.GT.MEDX) WRITE(6,903) MP,MEX
575 903 FORMAT (' ***** PLATE #',I3,' HAS ',I3,' EDGES.',
576 2' PROGRAM ABORTS SINCE MAX. EDGE DIMENSION IS EXCEEDED.',
577 2,' *****')
578 IF(MEX.GT.MEDX) STOP
579 DO 5 ME=1,MEX
580 C$$$
581 C$$$ XX(MP,ME,N)=X,Y,Z COMPONENTS OF CORNER #ME OF PLATE #MP.
582 C$$$ N=1(X),N=2(Y),N=3(Z). INPUT CORNER DATA AS FOLLOWS:
583 C$$$ 1.,1.,0.
584 C$$$ -1.,1.,0.
585 C$$$ -1.,-1.,0.
586 C$$$ 1.,-1.,0.
587 C$$$ THIS IS THE INPUT FOR A 2 METER SQUARE PLATE.
588 C$$$ NOTE THAT IF THERE IS MORE THAN ONE PLATE, THEN THE CORNER
589 C$$$ DATA FOR EACH PLATE WOULD FOLLOW SEQUENTIALLY.
590 C$$$
591 READ(5,*) (XX(MP,ME,N),N=1,3)

```

```

592 5      CONTINUE
593      WRITE(6,3302)(LABEL(J,IUNIT),J=1,2)
594 3302    FORMAT(2H *,2X,'PLATE#',2X,'CORNER#',3X,'INPUT LOCATION IN ',
595          12A3,4X,'ACTUAL LOCATION IN METERS',T79,1H*)
596      WRITE(6,3303)
597 3303    FORMAT(2H *,2X,'-----',2X,'-----'
598          1,2(2X,2('-----')),T79,1H*)
599      DO 3304 ME=1,MEX
600      WRITE(6,3306)
601      DO 3310 N=1,3
602 3310    XO(N)=XX(MP,ME,N)
603      DO 3311 N=1,3
604 3311    XX(MP,ME,N)=UNITS*(XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N))+TR(N)
605      WRITE(6,3305)MP,ME,(XO(N),N=1,3),(XX(MP,ME,N),N=1,3)
606 3305    FORMAT(2H *,4X,I3,6X,I2,2X,2(2X,F8.3,2(' ',F8.3)),T79,1H*)
607 3304    CONTINUE
608      GO TO 3000
609  C-----
610 3350    CONTINUE
611  C---  NP:  COMMAND  -----
612  C$$$
613  C$$$  INITIALIZE PLATE DATA.
614  C$$$
615      LPLA=.FALSE.
616      MPX=0
617      WRITE(6,3351)
618 3351    FORMAT(2H *,5X,' THE PLATE DATA IS INITIALIZED. ',T79,1H*/
619          22H *,5X,' NO PLATES ARE PRESENTLY IN THE PROBLEM. ',T79,1H*)
620      GO TO 3000
621  C-----
622 3400    CONTINUE
623  C---  SG:  COMMAND  -----
624  C$$$
625  C$$$  MSX=NUMBER OF ANTENNA ELEMENTS.
626  C$$$
627      LAMP=.FALSE.
628      MSX=MSX+1
629      IF (MSX.GT.MSDX) WRITE(6,904) MSX
630 904     FORMAT (' *****  NUMBER OF SOURCES= ',I3,' PROGRAM',
631          2' ABORTS SINCE MAX. SOURCE DIMENSION IS EXCEEDED.  *****')
632      IF (MSX.GT.MSDX) STOP
633      WRITE(6,3401) MSX
634 3401    FORMAT(2H *,5X,'THIS IS SOURCE NO. ',I3,' IN THIS',
635          1' COMPUTATION.',T79,1H*)
636      WRITE(6,3200)
637      WRITE(6,3006)
638  C$$$
639  C$$$  XSS(MS,N)=XYZ LOCATION OF MS-TH ANTENNA ELEMENT.
640  C$$$
641  C$$$  IMS(MS)=TYPE OF LINEAR ANTENNA
642  C$$$           0=ELECTRIC LINEAR ELEMENT
643  C$$$           1=MAGNETIC LINEAR ELEMENT
644  C$$$
645  C$$$  HAWS(MS)=APERTURE WIDTH IN WAVELENGTHS (NOTE: IF
646  C$$$           HAWS(MS) IS LESS THAN .1 LAMBDA, SOURCE IS
647  C$$$           CONSIDERED TO BE DIPOLE SOURCE)
648  C$$$  HS(MS)=LENGTH OF LINEAR ELEMENT IN WAVELENGTHS
649  C$$$
650  C$$$  THOZ(MS),PHOZ(MS)=ORIENTATION ANGLES USED TO DEFINE LINEAR
651  C$$$           ELEMENT AXIS.
652  C$$$
653  C$$$  THOX(MS),PHOX(MS)=ORIENTATION ANGLES USED TO DEFINE APERTURE
654  C$$$           PLANE OR DIPOLE X-AXIS.
655  C$$$
656  C$$$  EV(MS),EP(MS)=MAGNITUDE AND PHASE OF EXCITATION OF
657  C$$$           MS-TH ELEMENT.

```

```

058 CESS
059 MS=MSX
060 READ(5,*) (XSS(MS,N),N=1,3)
061 READ(5,*) THOZ(MS),PHOZ(MS),THOX(MS),PHOX(MS)
062 READ(5,*) IMS(MS),HS(MS),HAWS(MS)
063 READ(5,*) WM(MS),WP(MS)
064 IF(IMS(MS).EQ.0) WRITE(6,3411)
065 3411 FORMAT(2H *,5X,'THIS IS AN ELECTRIC SOURCE.',T79,1H*)
066 IF(IMS(MS).EQ.1) WRITE(6,3412)
067 3412 FORMAT(2H *,5X,'THIS IS A MAGNETIC SOURCE.',T79,1H*)
068 WRITE(6,3006)
069 IF(IUNST.EQ.0) GO TO 3414
070 UNSTS=UNIT(IUNST)
071 WRITE(6,3413) HS(MS),HAWS(MS),(LABEL(J,IUNST),J=1,2)
072 3413 FORMAT(2H *,5X,'SOURCE LENGTH=',F10.5,' AND WIDTH='
073 2,F10.5,1X,2A3,T79,1H*)
074 HS(MS)=UNSTS*HS(MS)
075 HAWS(MS)=UNSTS*HAWS(MS)
076 WRITE(6,3006)
077 WRITE(6,3413) HS(MS),HAWS(MS),(LABEL(J,1),J=1,2)
078 GO TO 3416
079 3414 WRITE(6,3415) HS(MS),HAWS(MS)
080 3415 FORMAT(2H *,5X,'SOURCE LENGTH=',F10.5,' AND WIDTH='
081 2,F10.5,' WAVELENGTHS',T79,1H*)
082 3416 WRITE(6,3006)
083 WRITE(6,3417) WM(MS),WP(MS)
084 3417 FORMAT(2H *,5X,'THE SOURCE WEIGHT HAS MAGNITUDE='
085 2,F10.5,' AND PHASE=',F10.5,T79,1H*)
086 WRITE(6,3006)
087 WRITE(6,3006)
088 WRITE(6,3421)(LABEL(J,IUNIT),J=1,2)
089 3421 FORMAT(2H *,T6,'SOURCE#',T17,'INPUT LOCATION IN ',2A3,T46,
090 1,'ACTUAL LOCATION IN METERS',T79,1H*)
091 WRITE(6,3422)
092 3422 FORMAT(2H *,T6,7(' '),T16,27(' '),T45,27(' '),
093 1,T79,1H*)
094 WRITE(6,3006)
095 DO 3424 N=1,3
096 3424 XO(N)=XSS(MS,N)
097 DO 3425 N=1,3
098 3425 XSS(MS,N)=UNITS*(XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N))+TR(N)
099 WRITE(6,3426)MS,(XO(N),N=1,3),(XSS(MS,N),N=1,3)
100 3426 FORMAT(2H *,T8,13,T15,F8.3,2(' ',F8.3),T44,F8.3,2(' ',F8.3)
101 1,T79,1H*)
102 TOR=THOZ(MS)*RPD
103 POR=PHOZ(MS)*RPD
104 XO(1)=SIN(TOR)*COS(POR)
105 XO(2)=SIN(TOR)*SIN(POR)
106 XO(3)=COS(TOR)
107 DO 3431 N=1,3
108 3431 VXSS(3,N,MS)=XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N)
109 TOR=THOX(MS)*RPD
110 POR=PHOX(MS)*RPD
111 XO(1)=SIN(TOR)*COS(POR)
112 XO(2)=SIN(TOR)*SIN(POR)
113 XO(3)=COS(TOR)
114 DO 3432 N=1,3
115 3432 VXSS(1,N,MS)=XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N)
116 DX=XVSS(1,1,MS)*VXSS(3,1,MS)+VXSS(1,2,MS)*VXSS(3,2,MS)
117 2+VXSS(1,3,MS)*VXSS(3,3,MS)
118 IF(ABS(DX).GT.0.1) WRITE(6,3430)
119 3430 FORMAT(' *** PROGRAM ABORTS IN SOURCE SECTION IN THAT THE
120 2' COORDINATES ARE NOT ORTHOGONAL !!! ***')
121 IF(ABS(DX).GT.0.1) STOP
122 VXSS(1,1,MS)=VXSS(1,1,MS)-VXSS(3,1,MS)*DX
123 VXSS(1,2,MS)=VXSS(1,2,MS)-VXSS(3,2,MS)*DX

```

```

724      VXSS(1,3,MS)=VXSS(1,3,MS)-VXSS(3,3,MS)*DZX
725      DOT=VXSS(1,1,MS)*VXSS(1,1,MS)+VXSS(1,2,MS)*VXSS(1,2,MS)
726      2*VXSS(1,3,MS)*VXSS(1,3,MS)
727      DOT=SQRT(DOT)
728      VXSS(1,1,MS)=VXSS(1,1,MS)/DOT
729      VXSS(1,2,MS)=VXSS(1,2,MS)/DOT
730      VXSS(1,3,MS)=VXSS(1,3,MS)/DOT
731      VXSS(2,1,MS)=VXSS(3,2,MS)*VXSS(1,3,MS)-VXSS(3,3,MS)*VXSS(1,2,MS)
732      VXSS(2,2,MS)=VXSS(3,3,MS)*VXSS(1,1,MS)-VXSS(3,1,MS)*VXSS(1,3,MS)
733      VXSS(2,3,MS)=VXSS(3,1,MS)*VXSS(1,2,MS)-VXSS(3,2,MS)*VXSS(1,1,MS)
734      WRITE(6,3006)
735      WHITE(6,3006)
736      WRITE(6,3437)
737 3437  FORMAT(2H *,5X,'THE FOLLOWING SOURCE ALIGNMENT IS USED'
738      2,T79,1H*)
739      DO 3433 NI=1,3
740      WHITE(6,3006)
741 3433  WHITE(6,3434) (NI,NJ,MS,VXSS(NI,NJ,MS),NJ=1,3)
742 3434  FORMAT(2H *,1X,3(2X,'VXSS(',11,',',11,',',12,',')=',F9.5)
743      2,T79,1H*)
744      GO TO 3000
745 C-----
746 3441  CONTINUE
747 C---  PH*  COMMAND  -----
748 C555
749 C555  PHAD=TOTAL POWER RADIATED IN WATTS.
750 C555
751 C555  PHAD CAN ALSO BE SPECIFIED AS THE POWER INPUT IN WATTS.
752 C555
753 C555  NOTE IF PHAD IS LESS THAN OR EQUAL TO 1.E-30
754 C555  THAN LPHAD WILL BE SET FALSE
755 C555
756      LPHAD=.TRUE.
757      READ(5,*) PHAD
758      IF (PHAD.LT.1.E-30) GO TO 3442
759      WHITE(6,3441) PHAD
760 3441  FORMAT(2H *,5X,'TOTAL POWER RADIATED IN WATTS= ',E12.6
761      2,T79,1H*)
762      GO TO 3000
763 3442  CONTINUE
764      LPHAD=.FALSE.
765      PHAD=0.
766      WHITE(6,3443)
767 3443  FORMAT(2H *,5X,'NO POWER RADIATED IS SPECIFIED',T79,1H*)
768      GO TO 3000
769 C-----
770 3450  CONTINUE
771 C---  AM*  COMMAND  -----
772 C555
773 C555  PHAD=TOTAL POWER RADIATED IN WATTS
774 C555
775      LPHAD=.TRUE.
776      READ(5,*) PHAD
777      WHITE(6,3441) PHAD
778      WHITE(6,3006)
779 C555
780 C555  MSX=NUMBER OF ANTENNA SEGMENTS
781 C555
782      LAMP=.TRUE.
783      READ(5,*) MSX
784      IF (MSX.GT.MSIX) WRITE(3,3477) MSX
785 3477  FORMAT(1X,10000  NUMBER OF SEGMENTS= ',13,
786      1X,' PROGRAM ABORTS SINCE MAX. SOURCE DIMENSION'
787      1X,' IS EXCEEDED.  ')
788      IF (MSX.GT.MSIX) STOP
789      WHITE(6,3451) MSX

```

```

700 1451 FORMAT(2H *,5X,'THERE ARE ',I3,' SEGMENTS IN THIS',
701 2' COMPUTATION.',T79,1H*)
702 WRITE(6,3006)
703 WRITE(6,3006)
704 C555
705 C555 XS(MS,N)=XYZ LOCATION OF MS-TH ANTENNA SEGMENT
706 C555
707 C555 IMS(MS)=0=ELECTRIC LINEAR ELEMENT
708 C555
709 C555 HS(MS)=LENGTH OF LINEAR ELEMENT
800 C555
801 C555 THOZ(MS),PHOZ(MS)=ORIENTATION ANGLES USED TO DEFINE
802 C555 LINEAR ELEMENT AXIS.
803 C555
804 C555 WR(MS),WP(MS)=REAL AND IMAGINARY CURRENT WEIGHT.
805 C555
806 WRITE(6,3006)
807 WRITE(6,3454)
808 1454 FORMAT(2H *,T31,'SEGMENT COORDINATES',T79,'*')
809 WRITE(6,3006)
810 WRITE(6,3006)
811 WRITE(6,3450) (LABEL(J,UNIT),J=1,2)
812 1450 FORMAT(2H *,T7,'MS',T14,'INPUT LOCATION IN ',2A3,
813 2T43,'ACTUAL LOCATION IN METERS',T79,'*')
814 WRITE(6,3457)
815 1457 FORMAT(2H *,T6,3(' '),T10,26(' '),T42,27(' '),T79,'*')
816 WRITE(6,3006)
817 DO 1452 MS=1,MSX
818 1452 IAS(MS)=0
819 DO 1119 MS=1,MSX
820 1119 HANS(MS)=0.
821 DO 1453 MS=1,MSX
822 1453 HEAD(5,*) (XSS(MS,N),N=1,3),HS(MS),THOZ(MS),PHOZ(MS)
823 HEAD(5,*) (WM(MS),WP(MS),MS=1,MSX)
824 DO 1473 MS=1,MSX
825 DO 1474 N=1,3
826 1474 XO(N)=X(S(MS,N))
827 DO 1475 N=1,3
828 1475 XSS(MS,N)=UNITS*(XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N)+TR(N))
829 WRITE(6,3470) MS,(XO(N),N=1,3),(XSS(MS,N),N=1,3)
830 1470 FORMAT(2H *,T6,I3,T13,FR.3,2(' ',FR.3),T42,FR.3,
831 2T(' ',FR.3),T79,1H*)
832 1473 CONTINUE
833 WRITE(6,3006)
834 WRITE(6,3006)
835 WRITE(6,3458) (LABEL(J,UNIT),J=1,2)
836 1458 FORMAT(2H *,T7,'MS',T13,'HS',2A3,T23,'MS=METERS',
837 2T41,'INPUT: THO,PHO',T43,'ACTUAL: THO,PHO',T79,1H*)
838 WRITE(6,3450)
839 1459 FORMAT(2H *,T6,3(' '),T12,20(' '),T40,16(' '),T79,
840 2T(' '),T79,1H*)
841 WRITE(6,3006)
842 DO 1403 MS=1,MSX
843 HSO=HS(MS)
844 HS(MS)=UNITS*HSO
845 TO=PI-THOZ(MS)
846 PO=PHOZ(MS)
847 XO(1)=SIN(TO*HPO)*COS(PO*HPO)
848 XO(2)=SIN(TO*HPO)*SIN(PO*HPO)
849 XO(3)=COS(TO*HPO)
850 DO 1401 N=1,3
851 1401 XQ(N)=XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N)
852 THOZ(MS)=ATAN2(SQRT(XQ(1)**2+XQ(2)**2),XQ(3))
853 PHOZ(MS)=ATAN2(XQ(2),XQ(1))
854 WRITE(6,3464) MS,HSO,HS(MS),TO,PO,THOZ(MS),PHOZ(MS)
855 1464 FORMAT(2H *,T6,I3,3X,2(2X,FR.4),5X,2(2X,FR.3),(' ',FR.3)

```

```

850      2,T79,1H*)
851      DO 3484 N=1,3
852 3484 VXSS(3,N,MS)=XOR(N)
853      VXSS(1,1,MS)=COS(THOZ(MS)*RPD)*COS(PHOZ(MS)*RPD)
854      VXSS(1,2,MS)=COS(THOZ(MS)*RPD)*SIN(PHOZ(MS)*RPD)
855      VXSS(1,3,MS)=-SIN(THOZ(MS)*RPD)
856      VXSS(2,1,MS)=-SIN(PHOZ(MS)*RPD)
857      VXSS(2,2,MS)=COS(PHOZ(MS)*RPD)
858      VXSS(2,3,MS)=0.
859 3463 CONTINUE
860      WRITE(6,3006)
861      WRITE(6,3006)
862      WRITE(6,3485)
863 3485 FORMAT(2H *,T33,'CURRENT WEIGHTS',T79,1H*,/2H *,T7,'MS',T18,
864      2'REAL',T31,'MAG.',T46,'MAG.',T57,'PHASE',T79,1H*)
865      WRITE(6,3486)
866 3486 FORMAT(2H *,T6,3(' '),T17,6(' '),T30,7(' '),T45,6(' '),
867      2T56,7(' '),T79,1H*)
868      DO 3465 MS=1,MSX
869      WMM=BABS(CMPLX(WM(MS),WP(MS)))
870      WPP=DPH*DTAN2(WP(MS),WM(MS))
871      WRITE(6,3466) MS,WMM(MS),WP(MS),WMM,WPP
872 3466 FORMAT(2H *,T6,I3.5X,E11.4,2X,E11.4,4X,E11.4,2X,FR.3,T79,1H*)
873 3465 CONTINUE
874      WRITE(6,3006)
875      GO TO 3000
876 C-----
877 3460 CONTINUE
878 C--- NS:  COMMAND -----
879 C555
880 C555 INITIALIZE SOURCE DATA.
881 C555
882      LAMP=.FALSE.
883      MSX=0
884      WRITE(6,3491)
885 3491 FORMAT(2H *.5X,' THE SOURCE DATA IS INITIALIZED. ',T79,1H*/
886      22H *.5X,' NO SOURCES ARE PRESENTLY IN THE PROBLEM. '
887      2,T79,1H*)
888      GO TO 3000
889 C-----
890 3500 CONTINUE
891 C--- LP:  COMMAND -----
892 C555
893 C555 LWRITE=TRUE IF LINE PRINTER OUTPUT OF DATA IS DESIRED
894 C555
895      READ(9,*) LWRITE
896      IF(.NOT.LWRITE) WRITE(6,3501)
897 3501 FORMAT(2H *.5X,'NO LINE PRINTER OUTPUT',T79,1H*)
898      IF(.NOT.LWRITE) GO TO 3000
899      WRITE(6,3501)
900 3501 FORMAT(2H *.5X,' DATA WILL BE OUTPUT ON LINE PRINTER !!!',
901      T79,1H*)
902      GO TO 3000
903 C-----
904 3600 CONTINUE
905 C--- PP:  COMMAND -----
906 C555
907 C555 LPLT=TRUE IF PEN PLOTTER OUTPUT IS DESIRED
908 C555
909      READ(9,*) LPLT
910      IF(.NOT.LPLT) WRITE(6,3601)
911 3601 FORMAT(2H *.5X,'NO PEN PLOT DESIRED',T79,1H*)
912      IF(.NOT.LPLT) GO TO 3000
913      IF LPLT=TRUE READ IN DIRECTIONS
914 C555

```

```

V22 C555 RADIUS=RADIUS OF POLAR PLOT.
V23 C555 IPLT=(FIELD PLOT), 2(POWER PLOT), 3(OB. PLOT)
V24 C555
V25      READ(5,*) RADIUS,IPLT
V26      WRITE(6,3002)
V27 3002 FORMAT(2H *.5X,'DATA WILL BE OUTPUT TO PEN PLOTTER !!!'
V28      2,T79,1H*)
V29      WRITE(6,3006)
V30      WRITE(6,3001)RADIUS,IPLT
V31 3001 FORMA.(2H *.5X,' RADIUS=',F6.2,5X,'IPLT=',I3,T79,1H*)
V32      GO TO 3000
V33 C-----
V34 3701 CONTINUE
V35 C--- GP:  COMMAND -----
V36 C555
V37 C555 INFINITE GROUND PLANE EFFECT INCLUDED.
V38 C555
V39      LORND=.TRUE.
V40      DO 3702 N=1,3
V41      XX(14,1,N)=1.E5*(XP(N)+YP(N))+TR(N)
V42      XX(14,2,N)=1.E5*(-XP(N)+YP(N))+TR(N)
V43      XX(14,3,N)=1.E5*(-XP(N)-YP(N))+TR(N)
V44 3702 XX(14,4,N)=1.E5*(XP(N)-YP(N))+TR(N)
V45      WRITE(6,3701)
V46 3701 FORMAT(2H *.5X,'INFINITE GROUND PLANE INSERTED IN'
V47      1,' STRUCTURE !!!',T79,1H*)
V48      WRITE(6,3006)
V49      WRITE(6,3703) (TR(N),N=1,3)
V50 3703 FORMAT(2H *.5X,'THE ORIGIN IS AT ',F12.6,',',F12.6
V51      2,',',F12.6,' METERS',T79,1H*)
V52      WRITE(6,3006)
V53      WRITE(6,3704) (ZP(N),N=1,3)
V54 3704 FORMAT(2H *.5X,'THE NORMAL IS ',F12.6,',',F12.6,',',
V55      2,F12.6,T79,1H*)
V56      GO TO 3000
V57 C-----
V58 3750 CONTINUE
V59 C--- NG:  COMMAND -----
V60 C555
V61 C555 INITIALIZE GROUND PLANE DATA.
V62 C555
V63      LORND=.FALSE.
V64      WRITE(6,3751)
V65 3751 FORMAT(2H *.5X,' GROUND PLANE DATA IS INITIALIZED. ',T79,1H*)
V66      22H *.5X,' NO GROUND PLANE IS PRESENTLY IN THE PROBLEM. '
V67      2,T79,1H*)
V68      GO TO 3000
V69 C-----
V70 3000 CONTINUE
V71 C--- WT:  COMMAND -----
V72 C555
V73 C555 TRINI=LINEAR TRANSLATION OF COORDINATES FROM THE FIXED
V74 C555 COORDINATES WHICH IS ORIGINALLY SET UP BY OPERATOR.
V75 C555
V76      READ(5,*) (TRINI,N=1,3)
V77 3021 FORMAT(2H *.5X,' INPUT DATA GIVEN IN TERMS OF ',F23.7,T79,1H*)
V78      WRITE(6,3001) (TRINI,N=1,3)
V79 3001 FORMAT(2H *.5X,' TRANSLATION IN ',F23.7,' TR(1)=',F8.3,
V80      1,' TR(2)=',F8.3,' TR(3)=',F8.3,T79,1H*)
V81      DO 3020 N=1,3
V82 3020 TRINI=TRINI+UNIT5
V83      WRITE(6,3001)
V84      IF (UNIT5.NE.1) WRITE(6,1001) LABEL(1,1),J=1,3), (TRINI,N=1,3)
V85      IF (UNIT5.NE.1) WRITE(6,1001)
V86      3001
V87 C111

```

```

986 C155 THZP,PHZP=ORIENTATION OF THE ZP AXIS RELATIVE TO THE
989 C155 FIXED COORDINATE SYSTEM.
990 C155
991 C155 THXP,PHXP=ORIENTATION OF THE XP AXIS RELATIVE TO THE
992 C155 FIXED COORDINATE SYSTEM.
993 C155
994 READ(5,*)THZP,PHZP,THXP,PHXP
995 ZP(1)=SIN(THZP*RPD)*COS(PHZP*RPD)
996 ZP(2)=SIN(THZP*RPD)*SIN(PHZP*RPD)
997 ZP(3)=COS(THZP*RPD)
998 XP(1)=SIN(THXP*RPD)*COS(PHXP*RPD)
999 XP(2)=SIN(THXP*RPD)*SIN(PHXP*RPD)
1000 XP(3)=COS(THXP*RPD)
1001 C111 INSURE XP IS PERPENDICULAR TO ZP
1002 DZX=ZP(1)*XP(1)+ZP(2)*XP(2)+ZP(3)*XP(3)
1003 IF(ABS(DZX).GT.0.1)WRITE(6,3003)
1004 3003 FORMAT(' *** PROGRAM ABORTS IN ROTATE SECTION IN THAT THE',
1005 ' COORDINATES ARE NOT ORTHOGONAL!!! ***')
1006 IF(ABS(DZX).GT.0.1)STOP
1007 XP(1)=XP(1)-ZP(1)*DZX
1008 XP(2)=XP(2)-ZP(2)*DZX
1009 DOT=SQRT(DOT)
1010 XP(3)=XP(3)-ZP(3)*DZX
1011 DOT=XP(1)*XP(1)+XP(2)*XP(2)+XP(3)*XP(3)
1012 XP(1)=XP(1)/DOT
1013 XP(2)=XP(2)/DOT
1014 XP(3)=XP(3)/DOT
1015 YP(1)=ZP(2)*XP(3)-ZP(3)*XP(2)
1016 YP(2)=ZP(3)*XP(1)-ZP(1)*XP(3)
1017 YP(3)=ZP(1)*XP(2)-ZP(2)*XP(1)
1018 WRITE(6,3001)
1019 3001 FORMAT(2H *,5X,'THE FOLLOWING ROTATIONS ARE USED FOR ALL',
1020 ' SUBSEQUENT INPUTS',T79,1H*)
1021 WRITE(6,3005)
1022 3005 WRITE(6,3002)(XP(N),N=1,3)
1023 3002 FORMAT(2H *,5X,'XP(1)=',F10.5,' XP(2)=',F10.5,' XP(3)=',
1024 'F10.5,T79,1H*)
1025 3005 WRITE(6,3003)(YP(N),N=1,3)
1026 3003 FORMAT(2H *,5X,'YP(1)=',F10.5,' YP(2)=',F10.5,' YP(3)=',
1027 'F10.5,T79,1H*)
1028 3005 WRITE(6,3004)
1029 3004 WRITE(6,3006)(ZP(N),N=1,3)
1030 3006 FORMAT(2H *,5X,'ZP(1)=',F10.5,' ZP(2)=',F10.5,' ZP(3)=',
1031 'F10.5,T79,1H*)
1032 3005 GO TO 3000
1033
1034 C-----
1035 3007 CONTINUE
1036 3007 301 CONNAIC -----
1037 C155
1038 C155 CYLINDER GEOMETRY INPUT
1039 C155
1040 3008 LEVE=TRUE.
1041 C155
1042 C155 APPROXIMATE OF ELLIPSE ON X CYLINDER AXIS
1043 C155 APPROXIMATE OF ELLIPSE ON Y CYLINDER AXIS
1044 C155
1045 C155 ZEN,THZ=MOST NEGATIVE END CAP'S Z COMPONENT
1046 C155 AND ANGLE OF SURFACE WITH THE NEG. CYLINDER AXIS
1047 C155 ZUP,THUP=MOST POSITIVE END CAP'S Z COMPONENT
1048 C155 AND ANGLE OF SURFACE WITH THE POS. CYLINDER AXIS
1049 C155
1050 READ(5,*)ZA,50
1051 READ(5,*)ZEN,THZ,ZUP,THUP
1052 FROM ZA
1053 TO 50

```

```

1054 ZCNO=ZCN
1055 ZCPO=ZCP
1056 AA=AA*UNITS
1057 BB=BB*UNITS
1058 ZCN=ZCN*UNITS
1059 ZCP=ZCP*UNITS
1060 WRITE(6,6310)(LABEL(J,IUNIT),J=1,2),AAO
1061 C310 FORMAT(2H *,5X,'X AXIS DIMENSION IN ',
1062 2A3,'"',F8.3,T79,1H*)
1063 WRITE(6,6006)
1064 WRITE(6,6310)(LABEL(J,1),J=1,2),AA
1065 WRITE(6,6006)
1066 WRITE(6,6006)
1067 WRITE(6,6320)(LABEL(J,IUNIT),J=1,2),BB0
1068 C320 FORMAT(2H *,5X,'Y AXIS DIMENSION IN ',
1069 2A3,'"',F8.3,T79,1H*)
1070 WRITE(6,6006)
1071 WRITE(6,6320)(LABEL(J,1),J=1,2),BB
1072 WRITE(6,6006)
1073 WRITE(6,6006)
1074 WRITE(6,6330)(LABEL(J,IUNIT),J=1,2),ZCNO
1075 C330 FORMAT(2H *,5X,'MOST NEGATIVE END CAP Z COMPONENT IN ',
1076 2A3,'"',F8.3,T79,1H*)
1077 WRITE(6,6006)
1078 WRITE(6,6330)(LABEL(J,1),J=1,2),ZCN
1079 WRITE(6,6006)
1080 WRITE(6,6340) THN
1081 C340 FORMAT(2H *,5X,'ANGLE OF NEG. END CAP SURFACE WITH NEG.',
1082 2' CYL. AXIS ',F8.3,T79,1H*)
1083 WRITE(6,6006)
1084 WRITE(6,6006)
1085 WRITE(6,6350)(LABEL(J,IUNIT),J=1,2),ZCPO
1086 C350 FORMAT(2H *,5X,'MOST POSITIVE END CAP Z COMPONENT IN ',
1087 2A3,'"',F8.3,T79,1H*)
1088 WRITE(6,6006)
1089 WRITE(6,6350)(LABEL(J,1),J=1,2),ZCP
1090 WRITE(6,6006)
1091 WRITE(6,6360) THP
1092 C360 FORMAT(2H *,5X,'ANGLE OF POS. END CAP SURFACE WITH POS.',
1093 2' CYL. AXIS ',F8.3,T79,1H*)
1094 WRITE(6,6006)
1095 DO 6370 I=1,3
1096 XXC(I)=TR(I)
1097 YC(I)=YP(I)
1098 ZC(I)=ZP(I)
1099 C370 ZC(I)=ZP(I)
1100 GO TO 1100
1101 C-----
1102 6000 CONTINUE
1103 C--- NO COMMAND -----
1104 C111
1105 C333 INITIALIZE CYLINDER DATA.
1106 C333
1107 LCYL=FALSE.
1108 WRITE(6,6051)
1109 C051 FORMAT(2H *,5X,'CYLINDER DATA IS INITIALIZED.',T79,1H*)
1110 2H *,5X,'NO CYLINDER IS PRESENTLY IN THE PROBLEM.'
1111 2,T79,1H*)
1112 GO TO 1100
1113 C-----
1114 600 CONTINUE
1115 C--- END COMMAND -----
1116 C333
1117 C333 END PROGRAM
1118 C333
1119 WRITE(6,6000)

```

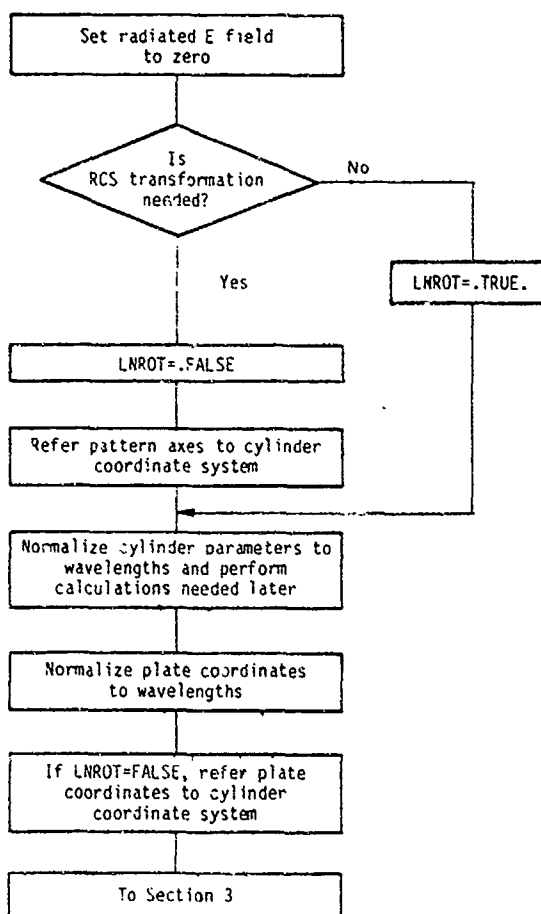
```

1120      WRITE(6,3006)
1121      WRITE(6,3005)
1122      GO TO 999
1123 C-----
1124 3800 CONTINUE
1125 C--- XO: COMMAND -----
1126 C$$$
1127 C$$$ EXECUTE PROGRAM
1128 C$$$
1129      WRITE(6,3006)
1130      WRITE(6,3006)
1131      WL=.2997925/FRQG
1132      WRITE(6,3005)
1133      MPXR=MPX
1134 C!!! GROUND PLANE IS ANOTHER PLATE IN SOLUTION.
1135      IF(LGRND)MPXR=MPX+1
1136      IF(MPXR.GT.MPDX)WRITE(6,901)MPXR
1137      IF(MPXR.GT.MPDX)GO TO 999
1138      IF(.NOT.LGRND)GO TO 3801
1139      LPLA=.TRUE.
1140      MEP(MPXH)=4
1141      DO 3802 I=1,4
1142      DO 3832 N=1,3
1143 3802 XX(MPXR,I,N)=WL*XX(MPDX,I,N)
1144 3801 CONTINUE
1145      IF(MPXR.EQ.0) LPLA=.FALSE.
1146 C!!!

```

2. Input Conversion Section

This section converts the input data into a preferred form for computational purposes. This involves converting angles in degrees to units of radians, linear dimensions into wavelengths, and performing the reference coordinate system (RCS) transformation if needed. The RCS transformation is done if a cylinder is present and its axis does not line up with the basic coordinate system used to define the input geometry.



```

1147 C!!! 2. INPUT CONVERSION SECTION
1148 C!!! NORMALIZE GEOMETRY UNITS (IN TERMS OF WAVELENGTHS) AND
1149 C!!! PERFORM RCS TRANSFORMATION (TO CYL COORD SYS) IF NEEDED
1150 C!!!
1151 C!!! SET E FIELDS TO ZERO
1152 DO 1 I=1,361
1153 ETHI(I)=(0.,0.)
1154 EPHT(I)=(0.,0.)
1155 FACTOR=1.
1156 BPL=0.
1157 SLR=BPL*RPD
1158 LNROT=.TRUE.
1159 DO 5101 N=1,3
1160 XPC(N)=XPD(N)
1161 YPC(N)=YPD(N)
1162 5101 ZPC(N)=ZPD(N)
1163 IF(.NOT.LCYL) GO TO 4
1164 LRFC=.FALSE.
1165 IF(.NOT.LPLA) GO TO 5106
1166 DO 5105 MP=1,MPX
1167 LRFI(MP)=.FALSE.
1168 LRFS(MP)=.FALSE.
1169 MEX=MEP(MP)
1170 DO 5105 ME=1,MEX
1171 LRDC(MP,ME)=.FALSE.
1172 5105 LRDC(MP,ME)=.FALSE.
1173 5106 CONTINUE
1174 C!!! DETERMINE IF RCS TRANSFORMATION IS NEEDED
1175 DO 8 N=1,3
1176 XCO(N)=XCO(N)/WL
1177 8 XCO(N)=0.
1178 LNROT=.TRUE.
1179 AXCL=ABS(XCL(1)-1.)
1180 AYCL=ABS(YCL(2)-1.)
1181 AZCL=ABS(ZCL(3)-1.)
1182 XCON=SQRT(XCO(1)*XCO(1)+XCO(2)*XCO(2)+XCO(3)*XCO(3))
1183 IF(AXCL.GT.1.E-5.OR.AYCL.GT.1.E-5) LNROT=.FALSE.
1184 IF(AZCL.GT.1.E-5.OR.XCON.GT.1.E-5) LNROT=.FALSE.
1185 IF(LNROT) GO TO 5108
1186 C!!! REFER PATTERN AXES TO CYL. AXES.
1187 CALL ROTRAN(XPC,XPD,XCO)
1188 CALL ROTRAN(YPC,YPD,XCO)
1189 CALL ROTRAN(ZPC,ZPD,XCO)
1190 5108 CONTINUE
1191 C!!! NORMALIZE CYLINDER COORDINATES
1192 A=AA/WL
1193 B=BB/WL
1194 ZC(1)=ZCP/WL
1195 ZC(2)=ZCN/WL
1196 THTPR=THTP*RPD
1197 SNC(1)=SIN(THTPR)
1198 CNC(1)=COS(THTPR)
1199 CTC(1)=CNC(1)/SNC(1)
1200 THTHR=THTH*RPD
1201 SNC(2)=SIN(THTHR)
1202 CNC(2)=COS(THTHR)
1203 CTC(2)=CNC(2)/SNC(2)
1204 4 CONTINUE
1205 C!!! NORMALIZE PLATE COORDINATES
1206 IF(.NOT.LPLA) GO TO 6
1207 DO 9 MP=1,MPX
1208 MEX=MEP(MP)
1209 DO 9 ME=1,MEX
1210 DO 9 N=1,3

```

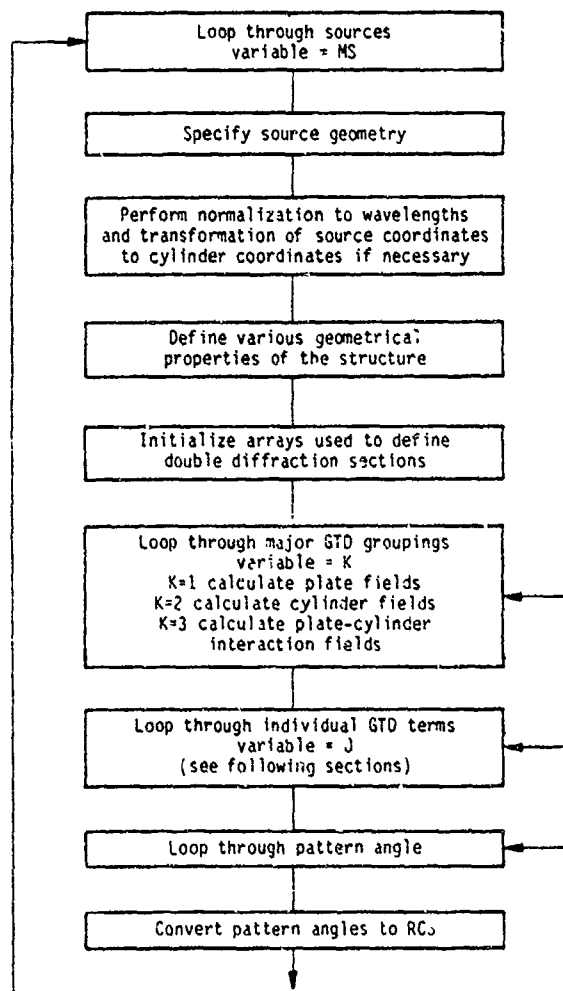
```

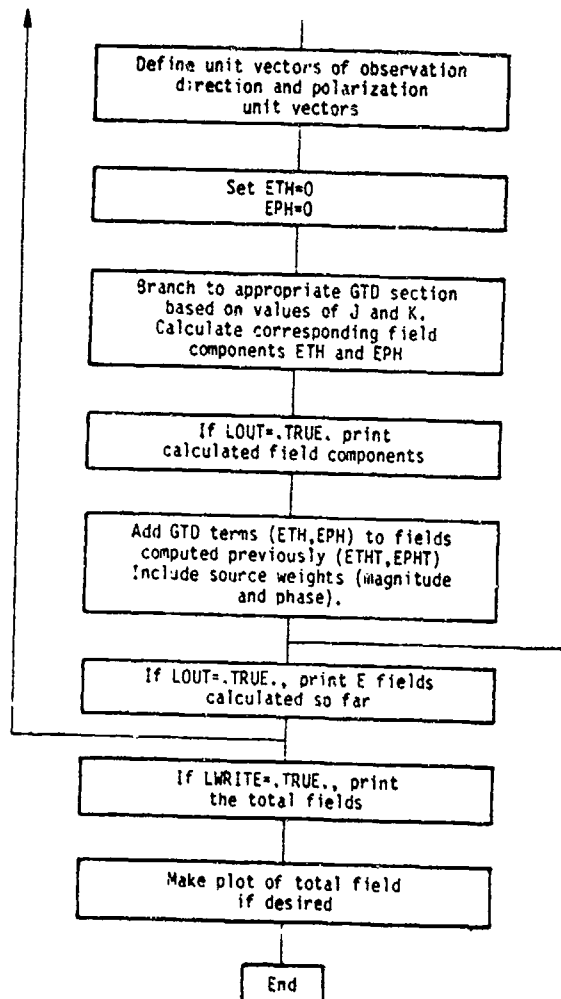
1211 S      X(MP,ME,N)=X(MP,ME,N)/ZL
1212        IF(LNRGT) GO TO 5220
1213        DO 5210 MP=1,MPXR
1214          MEX=MXP(MP)
1215          DO 5210 ME=1,MEX
1216            DO 5220 N=1,3
1217 5220      XXX(N)=X(MP,ME,N)
1218 C!!!     REFER PLATE COORD. TO CYL. COORD SYS
1219          CALL ROTRAN(XXX,XXX,XCO)
1220          DO 5230 N=1,3
1221 5230      X(MP,ME,N)=XXX(N)
1222 5210      CONTINUE
1223 5200      CONTINUE
1224 C        CONTINUE
1225 C!!!

```

3. Main Computation Section

This section directs the actual field calculations, performed in the various subroutines





```

1226 C!!! 3. MAIN COMPUTATION SECTION
1227 C!!!
1228 C!!! LOOP THRU VARIOUS SOURCES
1229 DO 1240 MS=1,MSX
1230 C!!! SPECIFY SOURCE GEOMETRY
1231 C!!! PERFORM NORMALIZATION AND TRANSFORMATION OF
1232 C!!! SOURCE COORDINATES
1233 DO 1235 N=1,3
1234 7 XS(N)=XS(MS,N)/WL
1235 IM=IMS(MS)
1236 DO 5307 NJ=1,3
1237 DO 5307 MI=1,3
1238 5307 VXS(MI,NJ)=VXSS(MI,NJ,MS)
1239 IF(LNROT) GO TO 5320
1240 C!!! REFER SOURCE LOCATION TO CYL. COORD SYS
1241 CALL ROTRAN(XS,XS,XCO)
1242 C!!! REFER SOURCE COORD SYS AXES TO CYL. COORD SYS
1243 DO 5304 MI=1,3
1244 DO 5303 NJ=1,3
1245 5303 XXX(NJ)=VXSS(MI,NJ,MS)
1246 CALL ROTRAN(XXX,XXX,XCO)
1247 DO 5304 NJ=1,3
1248 5304 VXS(MI,NJ)=XXX(NJ)
1249 5306 CONTINUE
1250 IF(LAMP) GO TO 5301
1251 IF(IMPST.NE.C) GO TO 5305
1252 C!!! SPECIFY SOURCE DIMENSIONS
1253 HAW=HAW(MS)
1254 H=HS(MS)
1255 GO TO 5306
1256 5305 HAW=HAW(MS)/WL
1257 H=HS(MS)/WL
1258 5306 HI=H(MS)*CEXP(CJ*WP(MS)*RFD)
1259 GO TO 5302
1260 C!!! SPECIFY SOURCE DIMENSIONS FOR REC INPUT
1261 5301 H=HS(MS)/WL
1262 HAW=0.
1263 HI=CMPLX(HAW(MS),WP(MS))
1264 IF(H.LT.0.15) HI=0.5*PI*HI
1265 5302 CONTINUE
1266 C!!! DEFINE VARIOUS GEOMETRY PROPERTIES OF STRUCTURE
1267 IF(LPLA) CALL GEOM
1268 IF(LCYL) CALL GEOMC
1269 IF(LPLA.AND.LCYL) CALL GEOMPC
1270 C!!!
1271 C!!! NOTE: AT THIS POINT THE RCS TRANSFORMATION (TO CYLINDER
1272 C!!! COORDINATES) IS COMPLETE. THE CYLINDER COORD SYS
1273 C!!! AND RCS ARE NOW THE SAME AND WILL BE REFERRED TO
1274 C!!! AS THE RCS (REFERENCE COORD SYS)
1275 C!!!
1276 C!!! INITIALIZE ARRAYS USED TO DEFINE DOUBLE DIFFRACTION SECTORS.
1277 DO 41 I=1,MEDX
1278 DO 41 J=1,MPDX
1279 41 ID(J,I)=-1
1280 DO 42 I=1,301
1281 42 IDD(I)=0
1282 KB=1
1283 KE=3
1284 IF(LSON) GO TO 1148
1285 IF(.NOT.LCYL) GO TO 1149
1286 IF(LPLA) GO TO 1149
1287 KB=2
1288 KE=2
1289 GO TO 1149
1290 1148 KB=1
1291 1149 KE=1

```

```

1292 1149 CONTINUE
1293 C!!! LOOP THRU MAJOR GTD GROUPS
1294 C!!! K=1 PLATE FIELDS
1295 C!!! K=2 CYLINDER FIELDS
1296 C!!! K=3 PLATE CYLINDER INTERACTION FIELDS
1297 DO 1150 K=Kb,Ke
1298 JB=Jb(K)
1299 JE=Jc(K)
1300 IF(LSOR) GO TO 1151
1301 IF(.NOT.LPLA.AND..NOT.LCYL) GO TO 1151
1302 IF(MPX.NE.0) GO TO 1152
1303 IF(K.EQ.2) GO TO 1152
1304 IF(JB.GT.2) GO TO 1150
1305 JE=2
1306 GO TO 1152
1307 1151 JB=1
1308 JE=1
1309 1152 CONTINUE
1310 IF(JB.EQ.0) GO TO 1150
1311 IF(JE.LT.JE) GO TO 1150
1312 C!!! LOOP THRU INDIVIDUAL GTD FIELDS.
1313 DO 1160 J=JB,JE
1314 C!!! LOOP THRU PATTERN ANGLE
1315 IF(LCNPA1) THP=TPPD
1316 IB=IB+1
1317 IEP=IEP+1
1318 IF(LDEBEG.OR.LTEST) IEP=IB+1
1319 DO 1160 II=IEP,IEP,IS
1320 C!!! CALCULATE PATTERN ANGLES IN PATTERN CUT COORD SYS.
1321 I=II-1
1322 PHP=I
1323 IF(LCNPA1) GO TO 1162
1324 IF(I.GT.180) GO TO 1161
1325 PHP=TPPD
1326 THP=I
1327 GO TO 1162
1328 1161 PHP=TPPD+180.
1329 IF(PHP.GE.360.) PHP=PHP-360.
1330 THP=360-I
1331 1162 THPR=THP*RPD
1332 PHPR=PHP*RPD
1333 C!!! CONVERT PATTERN ANGLES TO REF. COORD. SYS.
1334 CALL PATROT(THSR,PHSR,THPR,PHPR,ALR)
1335 STHS=SIN(THSR)
1336 CTHS=COS(THSR)
1337 SPS=SIN(PHSR)
1338 CPS=COS(PHSR)
1339 AS=PI-THSR
1340 SAS=SIN(AS)
1341 SASP=ABS(SIN(AS-0.5*PI))
1342 CAS=COS(AS)
1343 C!!! DEFINE OBSERVATION DIRECTION AND THETA,PHI UNIT VECTORS.
1344 D(1)=STHS*CPS
1345 D(2)=STHS*SPS
1346 D(3)=CTHS
1347 DT(1)=CTHS*CPS
1348 DT(2)=CTHS*SPS
1349 DT(3)=-STHS
1350 DP(1)=-SPS
1351 DP(2)=CPS
1352 ETP=(0.,0.)
1353 EPH=(0.,0.)
1354 C!!! BRANCH TO APPROPRIATE GTD SECTION BASED ON VALUES OF J AND K
1355 GO TO (116,112,113),K
1356 1110 CONTINUE
1357 GO TO (100,200,300,600,700,800,900),J

```

```

1356 100 CONTINUE
1359 C!!! COMPUTE THE DIRECT FIELD FROM THE SOURCE.
1360 CALL INCFD(EITH,EIPH,LSOR)
1361 ETH=EITH
1362 EPH=EIPH
1363 IF(LOUT) CALL PRIOUT(100,0,0,0,EITH,EIPH)
1364 GO TO 1000
1365 200 CONTINUE
1366 C!!! COMPUTE ALL POSSIBLE SINGLY REFLECTED FIELDS FROM PLATES.
1367 DO 25 MP=1,MPXR
1368 C!!! IF SLOT ON PLATE, THEN NO REFL. FIELD.
1369 IF(LSURF(MP)) GO TO 25
1370 C!!! IF PLATE SHADOWED, THEN NO REFL. FIELD.
1371 IF(LSHD(MP)) GO TO 25
1372 CALL REFPLA(ERPTH,ERPPH,MP)
1373 ETH=ETH+ERPTH
1374 EPH=EPH+ERPPH
1375 IF(LOUT) CALL PRIOUT(200,MP,0,0,ERPTH,ERPPH)
1376 25 CONTINUE
1377 GO TO 1000
1378 300 CONTINUE
1379 C!!! COMPUTE ALL POSSIBLE DOUBLY REFLECTED FIELDS.
1380 DO 31 MP=1,MPXR
1381 C!!! IF SLOT ON PLATE, THEN NO REFL/REFL FIELD.
1382 IF(LSURF(MP)) GO TO 31
1383 C!!! IF PLATE #MP IS SHADOWED, THEN NO REFL. FIELD
1384 IF (LSHD(MP)) GO TO 31
1385 DO 30 MPP=1,MPXR
1386 IF (MPP.EQ.MP) GO TO 30
1387 IF(LIHD(MP,MPP)) GO TO 30
1388 CALL RPLRPL(ERRPT,ERRPP,MP,MPP)
1389 ETH=ETH+ERRPT
1390 EPH=EPH+ERRPP
1391 IF(LOUT) CALL PRIOUT(300,MP,MPP,0,ERRPT,ERRPP)
1392 30 CONTINUE
1393 31 CONTINUE
1394 GO TO 1000
1395 600 CONTINUE
1396 C!!! COMPUTE ALL POSSIBLE SINGLY DIFFRACTED FIELDS INCLUDE
1397 C!!! A CORNER DIFFRACTION TERM IF DESIRED BY INPUT DATA.
1398 DO 61 MP=1,MPX
1399 C!!! IF PLATE SHADOWED, THEN NO DIFF. FIELD.
1400 IF(LSHD(MP)) GO TO 61
1401 MEX=MXP(MP)
1402 DO 60 ME=1,MEX
1403 FN=FNPM(MP,ME)
1404 C!!! IF WEDGE ANGLE INDICATOR (FN)<0, THEN HAVE COMMON EDGE ON
1405 C!!! OTHER PLATE COMPUTE DIFF. FIELD.
1406 IF(FN.LT.0.) GO TO 60
1407 CALL DIFPLT(EDPTH,EDPPH,EDPCTH,EDPCPH,FN,ME,MP)
1408 ETH=ETH+EDPTH+EDPCTH
1409 EPH=EPH+EDPPH+EDPCPH
1410 IF(LOUT) CALL PRIOUT(600,MP,ME,0,EDPTH,EDPPH)
1411 IF (LOUT) CALL PRIOUT(650,MP,ME,0,EDPCTH,EDPCPH)
1412 60 CONTINUE
1413 61 CONTINUE
1414 GO TO 1000
1415 100 CONTINUE
1416 C!!! LOOP THRU THE VARIOUS REFLECTED/DIFFRACTED FIELDS.
1417 C!!! INCLUDE CORNER TERM IF DESIRED BY INPUT DATA.
1418 DO 72 MR=1,MPXR
1419 C!!! IF SLOT ON PLATE, THEN NO REFL/DIFF FIELD.
1420 IF(LSURF(MR)) GO TO 72
1421 C!!! IF PLATE #MR IS SHADOWED, THEN NO REFL. FIELD
1422 IF (LSHD(MR)) GO TO 72
1423 DO 71 MP=1,MPX

```

```

1424 IF (MP.EC.NR) GO TO 71
1425 IF (LIND(MP,MP)) GO TO 71
1426 MEX=REP(MP)
1427 DO 70 ME=1,MEX
1428 FN=ERP(MP,ME)
1429 C!!! IF FN<0 THEN HAVE COMMON EDGE ON
1430 C!!! OTHER PLATE COMPUTE DIFF. FIELD.
1431 IF (FN.LT.0.) GO TO 70
1432 CALL RFLDPL(ERPDT,ERPDP,ERPCT,ERPDP,FN,ME,MP,NR)
1433 ETH=ETH+ERPDT+ERPCT
1434 EPH=EPH+ERPDP+ERPDP
1435 IF (LOUT) CALL PRIOUT(700,NR,MP,ME,ERPDT,ERPDP)
1436 IF (LOUT) CALL PRIOUT(750,NR,MP,ME,ERPCT,ERPDP)
1437 70 CONTINUE
1438 71 CONTINUE
1439 72 CONTINUE
1440 GO TO 1000
1441 800 CONTINUE
1442 C!!! COMPUTE THE VARIOUS DIFFRACTED/REFLECTED FIELDS.
1443 C!!! INCLUDE CORNER TERM IF DESIRED BY INPUT DATA.
1444 DO 82 MP=1,MPX
1445 C!!! IF PLATE IS SHADOWED, THEN NO DIFF/REFL FIELD.
1446 IF (LSD(MP)) GO TO 82
1447 MEX=REP(MP)
1448 DO 81 ME=1,MEX
1449 FN=ERP(MP,ME)
1450 IF (FN.LT.0.) GO TO 81
1451 DO 80 NR=1,MPXR
1452 IF (NR.EC.MP) GO TO 80
1453 IF (LIND(MP,ME)) GO TO 80
1454 CALL DPLRPL(EDRPT,EDRPP,EDRPT,EDRPP,FN,ME,MP,NR)
1455 ETH=ETH+EDRPT+EDRPT
1456 EPH=EPH+EDRPP+EDRPP
1457 IF (LOUT) CALL PRIOUT(800,NR,MP,ME,EDRPT,EDRPP)
1458 IF (LOUT) CALL PRIOUT(850,NR,MP,ME,NR,EDRPT,EDRPP)
1459 80 CONTINUE
1460 81 CONTINUE
1461 82 CONTINUE
1462 GO TO 1000
1463 900 CONTINUE
1464 C!!! CHECK TO SEE IF DOUBLE DIFFRACTIONS OCCUR.
1465 C!!! IF SO, INDICATE IN OUTPUT FILE.
1466 IF (ID(11).GE.0) GO TO 911
1467 ME=-ID(11)/400
1468 MP=-ID(11)/20-ME*20
1469 MPP=-ID(11)-ME*40-MP*20
1470 IF (LGCHL.AND.MPP.GE.MPX) GO TO 911
1471 IF (MP.EC.MP) GO TO 912
1472 WRITE(6,913) MP,ME,MPP
1473 913 FORMAT(' DOUBLE DIFFRACTION AT ANGLE= ',13,' FROM PLATE# ',
1474 2,12,' EDGE# ',12,' IS SHADOWED BY PLATE# ',12)
1475 GO TO 911
1476 912 WRITE(6,914) 1,MP,ME
1477 914 FORMAT(' DOUBLE DIFFRACTION AT ANGLE= ',13,' FROM PLATE# ',
1478 2,12,' EDGE# ',12,' IS SHADOWED BY THE CYLINDER')
1479 911 CONTINUE
1480 GO TO 1000
1481 1120 CONTINUE
1482 GO TO (101,100,500),J
1483 101 CONTINUE
1484 C!!! COMPUTE DIRECT FIELD FROM SOURCE
1485 IF (LPLA) GO TO 12
1486 CALL INFLD(ETH,EPH,LSO)
1487 ETH=ETH
1488 EPH=EPH
1489 IF (LOUT) GO TO 1000

```

```

1490 12      CONTINUE
1491 C!!!    COMPUTE SCATTERED FIELD FROM CYLINDER
1492          CALL SCTCYL(ESTH,ESPH,ERTH,ERPH)
1493          ETH=ETH+ESTH
1494          EPH=EPH+ESPH
1495          IF(.NOT.LOUT) GO TO 1496
1496          CALL PRIOUT(110,0,0,0,ETH,EIPH)
1497          CALL PRIOUT(120,0,0,0,ERTH,ERPH)
1498          CALL PRIOUT(130,0,0,0,ESTH,ESPH)
1499          GO TO 1490
1500 150      CONTINUE
1501 C!!!    COMPUTE ALL POSSIBLE REFLECTED FIELDS FROM END CAPS
1502          DO 15 MC=1,2
1503 C!!!    IF ANTENNA IS ON ENDCAP NO REFLECTED FIELD FROM END CAP
1504          IF(LSURFC(MC)) GO TO 15
1505          CALL REFCAP(ERCAT,ERCAP,MC)
1506          ETH=ETH+ERCAT
1507          EPH=EPH+ERCAP
1508          IF(LOUT) CALL PRIOUT(150,MC,0,0,ERCAT,ERCAP)
1509 15        CONTINUE
1510          GO TO 1490
1511 150      CONTINUE
1512 C!!!    COMPUTE ALL POSSIBLE DIFFRACTED FIELDS FROM END CAPS
1513          DO 50 MC=1,2
1514          CALL ENDIF(EDCTH,EDCPH,MC)
1515          ETH=ETH+EDCTH
1516          EPH=EPH+EDCPH
1517          IF(LOUT) CALL PRIOUT(500,MC,0,0,EDCTH,EDCPH)
1518 50        CONTINUE
1519          GO TO 1490
1520 1130     CONTINUE
1521          GO TO (250,400,940,950),J
1522 250      CONTINUE
1523 C!!!    COMPUTE ALL POSSIBLE FIELDS REFLECTED FROM THE PLATES THEN
1524 C!!!    SCATTERED FROM THE CYLINDER
1525          DO 29 MP=1,MPXR
1526 C!!!    IF ANTENNA IS ON PLATE, THEN NO REFLECTED FIELD
1527          IF(LSURF(MP)) GO TO 29
1528 C!!!    IF PLATE SHADOWED, THEN NO REFLECTED FIELD
1529          IF(LSHD(MP)) GO TO 29
1530          CALL RPLSCL(ERPST,ERPSP,ERPCT,ERPCP,MP)
1531          ETH=ETH+ERPST
1532          EPH=EPH+ERPSP
1533          IF(.NOT.LOUT) GO TO 29
1534          CALL PRIOUT(240,MP,0,0,ERPCT,ERPCP)
1535          CALL PRIOUT(250,MP,0,0,ERPST,ERPSP)
1536 29        CONTINUE
1537          GO TO 1490
1538 400      CONTINUE
1539 C!!!    COMPUTE ALL POSSIBLE FIELDS SCATTERED FROM THE CYLINDER THEN
1540 C!!!    REFLECTED FROM THE PLATES
1541          DO 40 MP=1,MPXR
1542          CALL SCLRPL(ERSPT,ERSPP,ERCPT,ERCPP,MP)
1543          ETH=ETH+ERSPT
1544          EPH=EPH+ERSPP
1545          IF(.NOT.LOUT) GO TO 40
1546          CALL PRIOUT(410,MP,0,0,ERCPT,ERCPP)
1547          CALL PRIOUT(420,MP,0,0,ERSPT,ERSPP)
1548 40        CONTINUE
1549          GO TO 1490
1550 940      CONTINUE
1551 C!!!    COMPUTE ALL POSSIBLE FIELDS REFLECTED FROM THE CYLINDER THEN
1552 C!!!    DIFFRACTED FROM THE PLATES
1553          DO 91 MP=1,MPX
1554 C!!!    IF PLATE SHADOWED, THEN NO DIFFRACTED FIELD
1555          IF(LSHD(MP)) GO TO 91

```

```

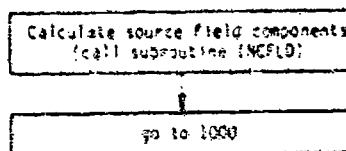
1556 MEX=MEP(MP)
1557 DO 96 ME=1,MEX
1558 FN=FNP(MP,ME)
1559 IF(FN.L1.5.) GO TO 96
1560 CALL RCLDPL(ERDTH,ERDPH,FN,ME,MP)
1561 ETH=ETH+ERDTH
1562 EPH=EPH+ERDPH
1563 IF(LOUT) CALL PRIOUT(941,MP,ME,0,ERDTH,ERDPH)
1564 96 CONTINUE
1565 91 CONTINUE
1566 GO TO 1200
1567 950 CONTINUE
1568 C!!! COMPUTE ALL POSSIBLE FIELDS DIFFRACTED FROM THE PLATES THEN
1569 C!!! REFLECTED FROM THE CYLINDER
1570 DO 96 MP=1,MTX
1571 C!!! IF PLATE SHADOWED, THEN NO DIFFRACTED FIELD
1572 IF(LSHD(MP)) GO TO 96
1573 MEX=MEP(MP)
1574 DO 95 ME=1,MEX
1575 C!!! IF EDGE DOES NOT HAVE STRONG FIELD REFLECTED FROM CYLINDER
1576 C!!! BYPASS SUBR.
1577 IF(.NOT.LDC(MP,ME)) GO TO 95
1578 FN=FNP(MP,ME)
1579 IF(FN.L1.5.) GO TO 95
1580 CALL DPLCCL(EDRCT,EDRCP,FN,ME,MP)
1581 ETH=ETH+EDRCT
1582 EPH=EPH+EDRCP
1583 IF(LOUT) CALL PRIOUT(950,MP,ME,0,EDRCT,EDRCP)
1584 95 CONTINUE
1585 96 CONTINUE
1586 1200 CONTINUE
1587 IF(LOUT) CALL PRIOUT(1,1,J,J,ETH,EPH)
1588 C!!! SUPERPOSITION OF THE FIELD COMPONENTS, WEIGHTING
1589 C!!! OF RESULT IN TERMS OF THE INPUT EXCITATION, AND
1590 C!!! THE CONVERSION OF THE POLARIZATION TO THE
1591 C!!! PATTERN CUT COORDINATE SYSTEM.
1592 ETH(II)=ETH(II)+II*(ETH*COS(ALR+BLR)+EPH*SIN(ALR+BLR))
1593 EPHT(II)=EPHT(II)+II*(EPH*COS(ALR+BLR)-ETH*SIN(ALR+BLR))
1594 1100 CONTINUE
1595 1150 CONTINUE
1596 IF (.NOT.LOUT) GO TO 1200
1597 DO 1202 II=IEP,IEP,IS
1598 I=II-1
1599 1202 CALL PRIOUT (1000,I,I,I,ETH(II),EPHT(II))
1600 1200 CONTINUE
1601 C!!! E-THETA AND E-PHI RESULTS ARE SENT TO UNIT #6(LINE PRINTER).
1602 IEE=IEP-1
1603 IF(LWRITE) CALL OUTPUT(ETH,EPHT,LCPAT,TPOD,13,IEE,IS)
1604 C!!! POLAR PLOT OF DATA IF DESIRED.
1605 C!!! NOTE THAT THE PLOT ROUTINES ARE NOT INCLUDED
1606 C!!! SINCE THEY CAN NOT BE USED ON ALL SYSTEMS.
1607 IF(.NOT.LPLT) GO TO 993
1608 C ADD CALL POLPLT(ETH,RADIUS,IPLT,IS,361)
1609 C ADD CALL POLPLT(EPHT,RADIUS,IPLT,IS,361)
1610 990 CONTINUE
1611 GO TO 2599
1612 999 STOP
1613 END

```

A description of the various GTD computation sections based on values of J and K follows. A partial listing of each section is repeated for clarity.

K=1, J=1

This section calculates the geometrical optics source field.

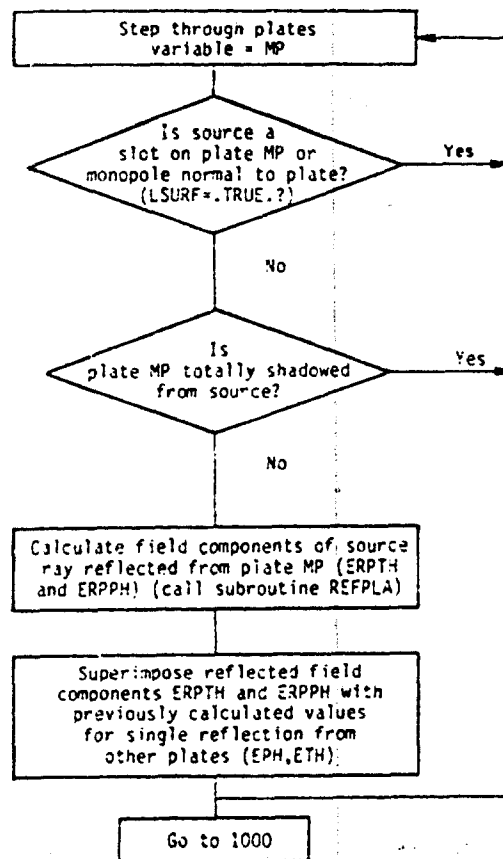


```

1350 122 CONTINUE
1351 1111 COMPUTE THE DIRECT FIELD FROM THE SOURCE.
1352 CALL INCFLO(EITH,EIPM,LSO)
1353 EIPM=EIP
1354 EPM=EIPM
1355 IF(OUT) CALL PRICOUT(122,0.0,0.0,EITH,EIPM)
1356 GO TO 1100
  
```

K=1,J=2

This section calculates all fields singly reflected from plates.

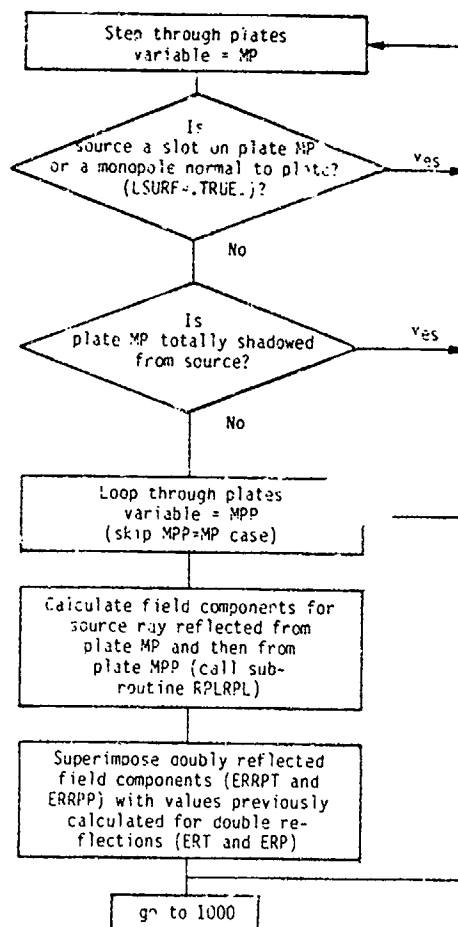


```

1365 200  CONTINUE
1366 C!!!  COMPUTE ALL POSSIBLE SINGLY REFLECTED FIELDS FROM PLATES.
1367      DO 25 MP=1,MPXR
1368 C!!!  IF SLOT ON PLATE, THEN NO REFL. FIELD.
1369      IF(LSURF(MP)) GO TO 25
1370 C!!!  IF PLATE SHADOWED, THEN NO REFL. FIELD.
1371      IF(LSHD(MP)) GO TO 25
1372      CALL REFPLA(ERPTH,ERPPH,MP)
1373      ETH=ETH+ERPTH
1374      EPH=EPH+ERPPH
1375      IF(LOUT) CALL PRIOUT(200,MP,0,0,ERPTH,ERPPH)
1376 25    CONTINUE
1377      GO TO 1000
  
```

K 1, J-3

This section computes all possible doubly reflected fields from plates.

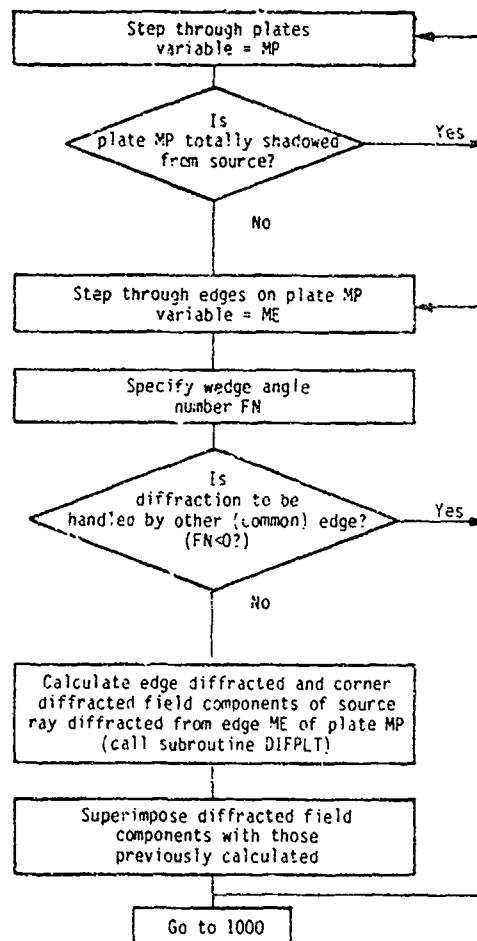


```

1378 300  CONTINUE
1379 C!!!  COMPUTE ALL POSSIBLE DOUBLY REFLECTED FIELDS.
1380      DO 31 MP=1,MPXR
1381 C!!!  IF SLOT ON PLATE, THEN NO REFL/REFL FIELD.
1382      IF (LSURF(MP)) GO TO 31
1383 C!!!  IF PLATE #MP IS SHADOWED, THEN NO REFL. FIELD
1384      IF (LSHD(MP)) GO TO 31
1385      DO 30 MPP=1,MPXR
1386      IF (MPP.EQ.MP) GO TO 30
1387      IF (LIHD(MP,MPP)) GO TO 30
1388      CALL RPLRPL(ERRPT,ERRPP,MP,MPP)
1389      ETH=ETH+ERRPT
1390      EPH=EPH+ERRPP
1391      IF (LOUT) CALL PRIOUT(300,MP,MPP,0,ERRPT,ERRPP)
1392 30  CONTINUE
1393 31  CONTINUE
1394      GO TO 1000
  
```

K=1, J=4

This section computes field components for all source rays singly diffracted by plate edges.



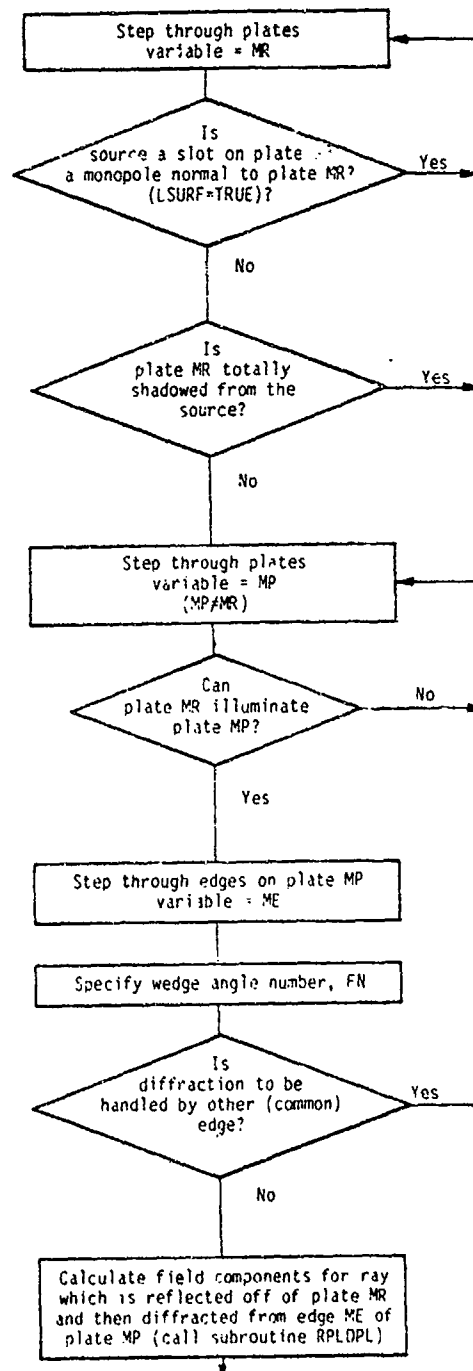
```

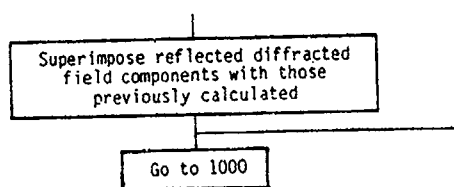
1395 C00 CONTINUE
1396 C!!! COMPUTE ALL POSSIBLE SINGLY DIFFRACTED FIELDS INCLUDE
1397 C!!! A CORNER DIFFRACTION TERM IF DESIRED BY INPUT DATA.
1398 DO 61 MP=1,MPX
1399 C!!! IF PLATE SHADOWED, THEN NO DIFF. FIELD.
1400 IF(LSHD(MP)) GO TO 61
1401 MEX=MED(MP)
1402 DO 60 ME=1,MEX
1403 FN=FNP(MP,ME)
1404 C!!! IF WEDGE ANGLE INDICATOR (FN)<0, THEN HAVE COMMON EDGE ON
1405 C!!! OTHER PLATE COMPUTE DIFF. FIELD.
1406 IF(FN.L1.0.) GO TO 60
1407 CALL DIFPLT(EDPTH,EDPPH,EDPCTH,EDPCPH,FN,ME,MP)
1408 ETH=ETH+EDPTH+EDPCTH
1409 EPH=EPH+EDPPH+EDPCPH
1410 IF(LOUT) CALL PRIOUT(600,MP,ME,0,EDPTH,EDPPH)
1411 IF(LOUT) CALL PRIOUT(650,MP,ME,0,EDPCTH,EDPCPH)
1412 C00 CONTINUE
1413 C1 CONTINUE
1414 GO TO 1000

```

K=1, J=5

This section computes field components for all source rays reflected by a plate and then diffracted from an edge on another plate.



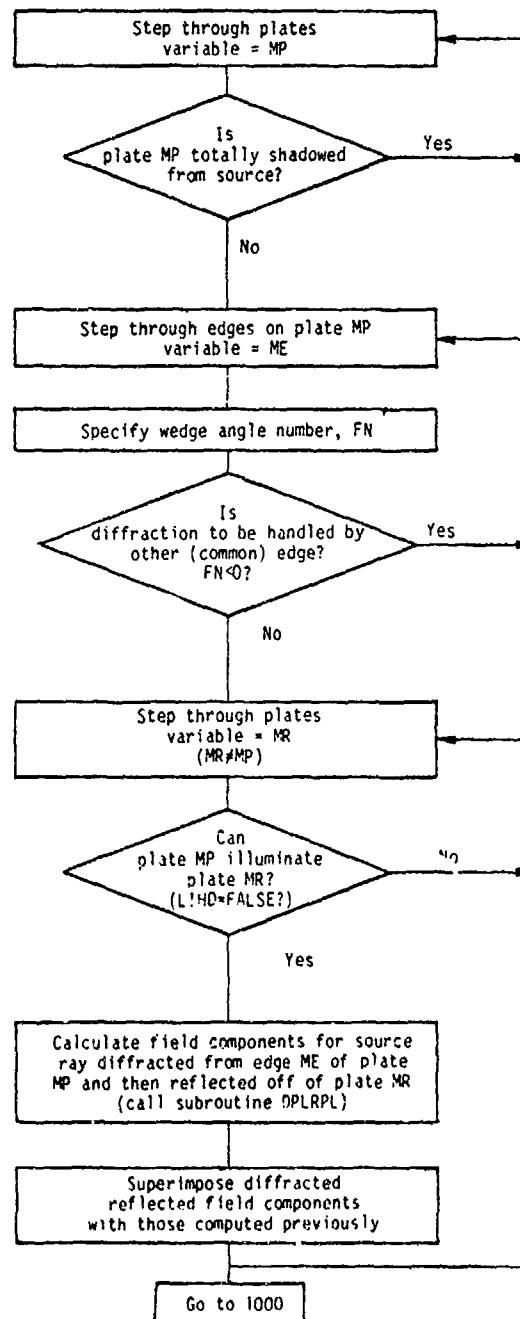


```

1415 700 CONTINUE
1416 C!!! LOOP THRU THE VARIOUS REFLECTED/DIFFRACTED FIELD TERMS.
1417 C!!! INCLUDE CORNER TERM IF DESIRED BY INPUT DATA.
1418 DO 72 MR=1,MPXR
1419 C!!! IF SLOT ON PLATE, THEN NO REFL/DIFF FIELD.
1420 IF(LSURF(MR)) GO TO 72
1421 C!!! IF PLATE #MR IS SHADOWED, THEN NO REFL. FIELD
1422 IF (LSHD(MR)) GO TO 72
1423 DO 71 MP=1,MPX
1424 IF (MP.EQ.MR) GO TO 71
1425 IF (LIHD(MR,MP)) GO TO 71
1426 MEX=NEP(MP)
1427 DO 70 ME=1,MEX
1428 FN=HP(MP,ME)
1429 C!!! IF FN<0 THEN HAVE COMMON EDGE ON
1430 C!!! OTHER PLATE COMPUTE DIFF. FIELD.
1431 IF(FN.L1.0.) GO TO 70
1432 CALL RPLDPL(ERPDT,ERPDP,ERPDCP,ERPDCP,FN,ME,MP,MR)
1433 ETH=ETH+ERPDT+ERPDCP
1434 EPH=EPH+ERPDP+ERPDCP
1435 IF(LOUT) CALL PRIOUT(700,MR,MP,ME,ERPDT,ERPDP)
1436 IF (LOUT) CALL PRIOUT(750,MR,MP,ME,ERPDCP,ERPDCP)
1437 70 CONTINUE
1438 71 CONTINUE
1439 72 CONTINUE
1440 GO TO 1000
  
```

K=1 J=6

This section computes field components for all source rays diffracted from a plate edge and then reflected off of another plate.



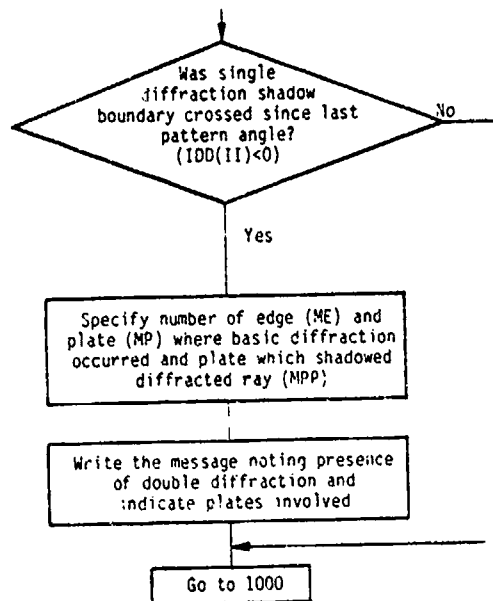
```

1441 800 CONTINUE
1442 C!!! COMPUTE THE VARIOUS DIFFRACTED/REFLECTED FIELDS.
1443 C!!! INCLUDE CORNER TERM IF DESIRED BY INPUT DATA.
1444 DO 82 MP=1,MPX
1445 C!!! IF PLATE IS SHADOWED, THEN NO DIFF/REFL FIELD.
1446 IF(LSHD(MP)) GO TO 82
1447 MEX=MED(MP)
1448 DO 81 ME=1,MEX
1449 FN=FN(MP,ME)
1450 IF(FN.LT.0.) GO TO 81
1451 DO 80 MR=1,MPXR
1452 IF(MR.EQ.MP) GO TO 80
1453 IF(LIHD(MP,MR)) GO TO 80
1454 CALL DPLRPL(EDRPT,EDRPP,EDCRPT,EDCRPP,FN,ME,MP,MR)
1455 ETH=ETH+EDRPT+EDCRPT
1456 EPH=EPH+EDRPP+EDCRPP
1457 IF(LOUT) CALL PRIOUT(800,MP,ME,MR,EDRPT,EDRPP)
1458 IF(LOUT) CALL PRIOUT(850,MP,ME,MR,EDCRPT,EDCRPP)
1459 80 CONTINUE
1460 81 CONTINUE
1461 82 CONTINUE
1462 GO TO 1600

```

K=1, J=7

This section identifies double diffraction shadow boundaries.

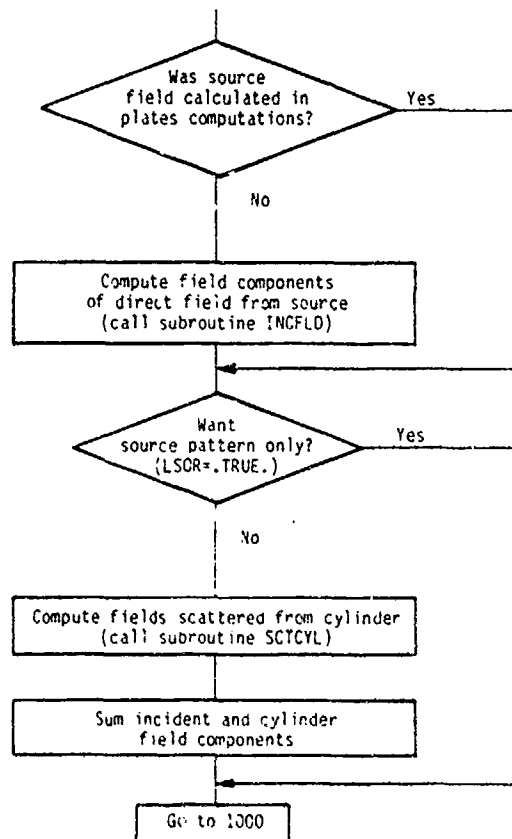


```

1403 500  CONTINUE
1404 C!!!  CHECK TO SEE IF DOUBLE DIFFRACTIONS OCCUR.
1405 C!!!  IF SO, INDICATE IN OUTPUT FILE.
1406      IF(IDD(11).GE.0)GO TO 911
1407      ME=-IDD(11)/400
1408      MP=-IDD(11)/20-ME*20
1409      MPP=-IDD(11)-ME*400-MP*20
1410      IF(LGRND.AND.MPP.GE.MPXR) GO TO 911
1411      IF(MPP.EQ.0) GO TO 912
1412      WRITE(6,913)1,MP,ME,MPP
1413 913    FORMAT(' DOUBLE DIFFRACTION AT ANGLE= ',I3,' FROM PLATE# ',
1414            2,I2,' EDGE# ',I2,' IS SHADOWED BY PLATE# ',I2)
1415      GO TO 911
1416 912    WRITE(6,914) 1,MP,ME
1417 914    FORMAT(' DOUBLE DIFFRACTION AT ANGLE= ',I3,' FROM PLATE# ',
1418            2,I2,' EDGE# ',I2,' IS SHADOWED BY THE CYLINDER')
1419 911    CONTINUE
1420      GO TO 1000
  
```

K=2, J=1

This section computes the source field and the field scattered from the cylinder.

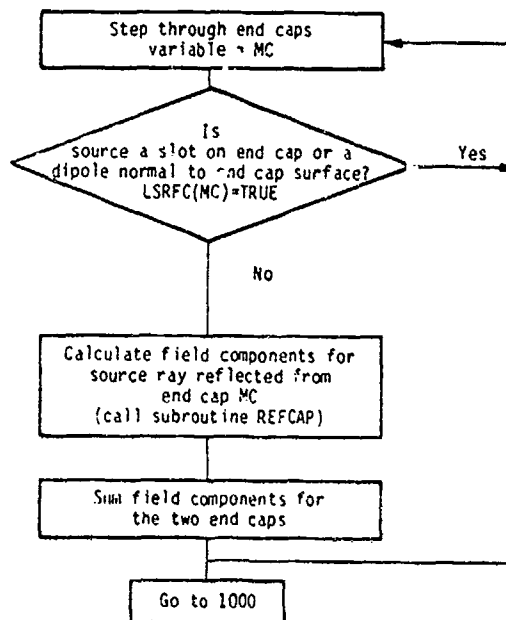


```

1483 101  CONTINUE
1484 C!!!  COMPUTE DIRECT FIELD FROM SOURCE
1485       IF(LPLA) GO TO 12
1486       CALL INCFLD(E1TH,E1PH,LSOR)
1487       E1H=E1TH
1488       E1P=E1PH
1489       IF(LSOR) GO TO 1000
1490 12    CONTINUE
1491 C!!!  COMPUTE SCATTERED FIELD FROM CYLINDER
1492       CALL SCTCYL(E1TH,E1PH,E2TH,E2PH)
1493       E1H=E1H+E2H
1494       E1P=E1P+E2P
1495       IF(.NOT.LOUT) GO TO 1000
1496       CALL PRIOUT(110,0,0,0,E1P,E1H)
1497       CALL PRIOUT(120,0,0,0,E2H,E2P)
1498       CALL PRIOUT(130,0,0,0,E2H,E2P)
1499       GO TO 1000
  
```

K=2, J=2

This section computes fields reflected from cylinder end caps.

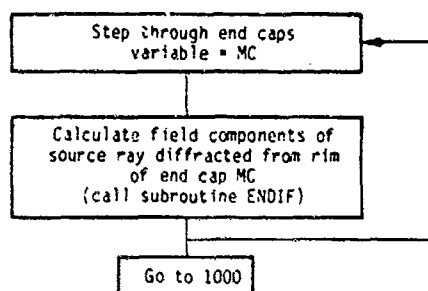


```

1500 150  CONTINUE
1501 C!!!  COMPUTE ALL POSSIBLE REFLECTED FIELDS FROM END CAPS
1502      DO 15 MC=1,2
1503 C!!!  IF ANTENNA IS ON ENDCAP NO REFLECTED FIELD FROM END CAP
1504      IF (LSRFC(MC)) GO TO 15
1505      CALL REFCAP(ERCAP,ERCAP,MC)
1506      ETH=ETH+ERCAP
1507      EPH=EPH+ERCAP
1508      IF (LOUT) CALL PRIOUT(150,MC,0,0,ERCAP,ERCAP)
1509 15  CONTINUE
1510  GO TO 1600
  
```

K=2, J=3

This section computes field components for all source rays diffracted from the cylinder end cap rims.

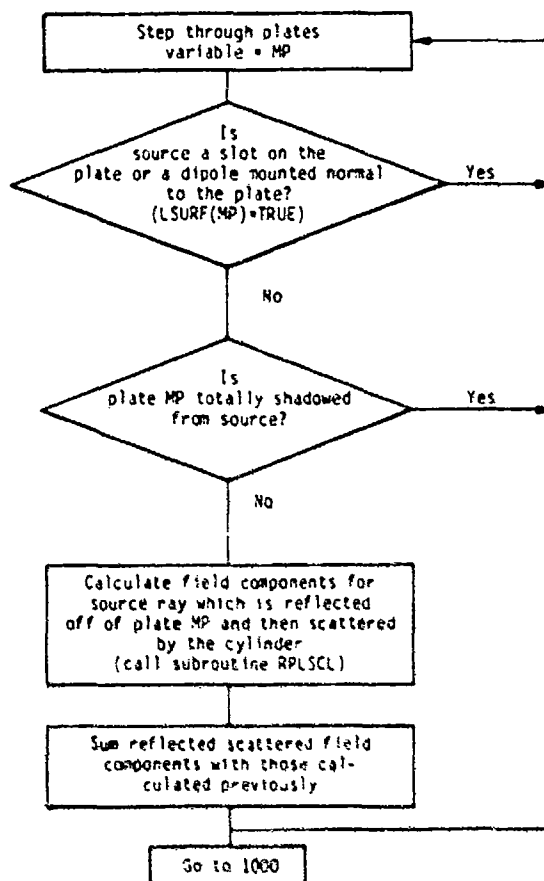


```

1511 SWO CONTINUE
1512 C111 COMPUTE ALL POSSIBLE DIFFRACTED FIELDS FROM END CAPS
1513 DO 50 MC=1,2
1514 CALL ENDIF(EDCTH,EDCPH,MC)
1515 ETH=ETH+EDCTH
1516 EPH=EPH+EDCPH
1517 IF(LOUT) CALL PRIOUT(SWJ,MC,0,0,EDCTH,EDCPH)
1518 SWO CONTINUE
1519 GO TO 1100
  
```

K=3, J=1

This section computes field components for all source rays which are reflected from a plate and then scattered by the cylinder.

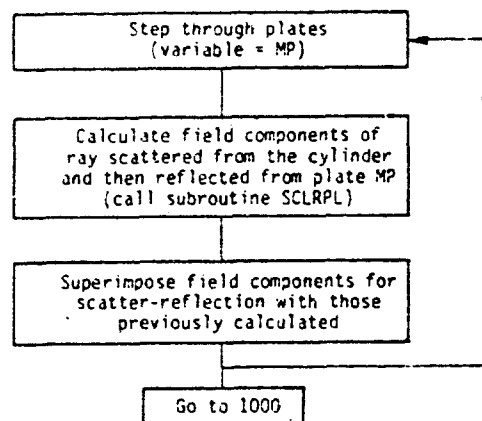


```

1522 250 CONTINUE
1523 C111 COMPUTE ALL POSSIBLE FIELDS REFLECTED FROM THE PLATES THEN
1524 C111 SCATTERED FROM THE CYLINDER
1525 DO 20 MP=1,MPXN
1526 C111 IF ANTENNA IS ON PLATE, THEN NO REFLECTED FIELD
1527 IF(LSURF(MP)) GO TO 20
1528 C111 IF PLATE SHADOWED, THEN NO REFLECTED FIELD
1529 IF(LSID(MP)) GO TO 20
1530 CALL RPLSCL(ENPST,ERPSP,ERPCT,ERPCP,MP)
1531 ETH=ETH+ERPST
1532 EPH=EPH+ERPSP
1533 IF(.NOT.LOUT) GO TO 20
1534 CALL PRIOUT(240,NO,0,0,ERPCT,ERPCP)
1535 CALL PRIOUT(250,"P",0,0,ENPST,ERPSP)
1536 20 CONTINUE
1537 GO TO 1000
  
```

K=3, J=2

This section calculates field components for all source rays scattered from the cylinder and then reflected from a plate.

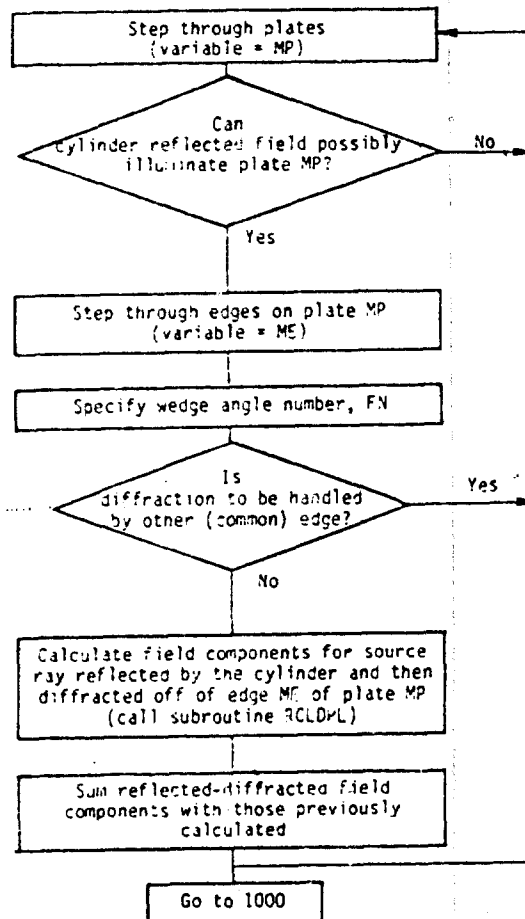


```

1538 400  CONTINUE
1539 C!!!  COMPUTE ALL POSSIBLE FIELDS SCATTERED FROM THE CYLINDER THEN
1540 C!!!  REFLECTED FROM THE PLATES
1541      DO 40 MP=1,MPXR
1542      CALL SCLRPL(ERSPT,ERSPP,ERCPT,ERCPP,MP)
1543      ETH=ETH+ERSPT
1544      EPH=EPH+ERSPP
1545      IF(.NOT.LOUT) GO TO 40
1546      CALL PRIOUT(410,MP,0,0,ERCPT,ERCPP)
1547      CALL PRIOUT(420,MP,0,0,ERSPT,ERSPP)
1548 40    CONTINUE
1549      GO TO 1000
  
```

K=3, J=3

This section computes field components for all source rays reflected from the cylinder and diffracted from a plate edge.

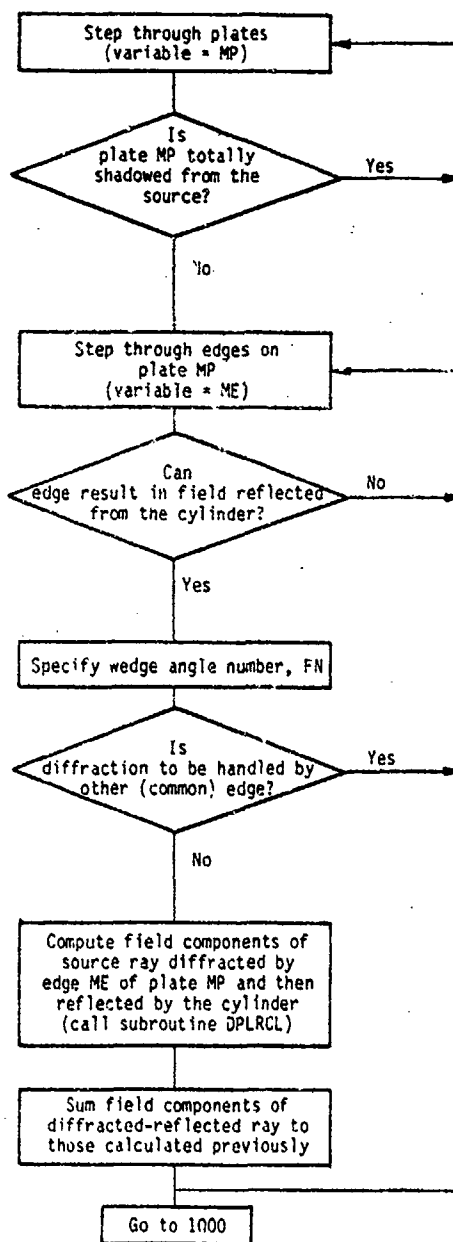


```

1550 940  CONTINUE
1551 C!!!  COMPUTE ALL POSSIBLE FIELDS REFLECTED FROM THE CYLINDER THEN
1552 C!!!  DIFFRACTED FROM THE PLATES
1553      DO 91 MP=1,MPX
1554 C!!!  IF PLATE SHADOWED, THEN NO DIFFRACTED FIELD
1555      IF(LSHD(MP)) GO TO 91
1556      MEX=MEP(MP)
1557      DO 90 ME=1,MEX
1558      FN=FN2(MP,ME)
1559      IF(FN.L1.0.) GO TO 90
1560      CALL RCLDPL(ERDTH,ERDPH,FN,ME,MP)
1561      ETH=ETH+ERDTH
1562      EPH=EPH+ERDPH
1563      IF(LOUT) CALL PRIOUT(940,MP,ME,0,ERDTH,ERDPH)
1564 90     CONTINUE
1565 91     CONTINUE
1566      GO TO 1000
  
```

K=3, J=4

This section computes field components of all source rays diffracted from plate edges and then reflected from the cylinder.



```

1567 950 CONTINUE
1568 C!!! COMPUTE ALL POSSIBLE FIELDS DIFFRACTED FROM THE PLATES THEN
1569 C!!! REFLECTED FROM THE CYLINDER
1570 DO 96 MP=1,MPX
1571 C!!! IF PLATE SHADOWED, THEN NO DIFFRACTED FIELD
1572 IF(LSID(MP)) GO TO 96
1573 MEX=MEP(MP)
1574 DO 95 ME=1,MEX
1575 C!!! IF EDGE DOES NOT HAVE STRONG FIELD REFLECTED FROM CYLINDER
1576 C!!! BYPASS SUBR.
1577 IF(.NOT.LDC(MP,ME)) GO TO 95
1578 FN=FNPM(MP,ME)
1579 IF(FN.LT.0.) GO TO 95
1580 CALL DPLRCL(EDRCT,EDRCP,FN,ME,MP)
1581 ETH=ETH+EDRCT
1582 EPH=EPH+EDRCP
1583 IF(I.OUT) CALL PRIOUT(950,MP,ME,0,EDRCT,EDRCP)

```

SYMBOL DICTIONARY

A	RADIUS OF CYLINDER ALONG X AXIS IN WAVELENGTHS
ALR	ANGLE THAT CONVERTS FIELD POLARIZATION FROM REFERENCE COORDINATE SYSTEM TO PATTERN CUT COORDINATE SYSTEM
AS	PI-THSR
AXCYL	VARIABLES USED TO DETERMINE IF THE CYLINDER COORDINATE SYSTEM IS THE SAME AS THE REFERENCE COORDINATE SYSTEM (BEFORE RCS TRANSFORMATION)
AYCYL	
AZCYL	
B	RADIUS OF CYLINDER ALONG Y AXIS IN WAVELENGTHS
BLR	BPL IN RADIAN
BPL	ANGLE THAT CONVERTS FIELD POLARIZATION FROM PATTERN CUT COORDINATE SYSTEM TO A RECEIVER COORDINATE SYSTEM (NOT PRESENTLY IMPLEMENTED)
CAS	COSINE OF AS
CNC	COSINE OF THTPR AND THTNR
CPS	COSINE OF PHSR
CTC	COTANGENT OF THTPR AND THTNR
CTHS	COSINE OF THSR
EDCPH	PHI COMPONENT OF FIELD DIFFRACTED FROM END CAP RIM IN RCS
EDCRPP	PHI COMPONENT OF FIELD DIFFRACTED FROM CORNERS OF EDGE ME OF PLATE MP AND THEN REFLECTED BY PLATE MR (CORNER DIF)
EDCTH	THETA COMPONENT OF FIELD DIFFRACTED FROM END CAP RIM IN RCS
EDCRPT	THETA COMPONENT OF FIELD DIFFRACTED FROM THE CORNERS OF EDGE ME OF PLATE MP AND THEN REFLECTED BY PLATE MR (CORNER DIFFRACTION)
EDPCPH	PHI COMPONENT OF FIELD DIFFRACTED FROM CORNERS OF EDGE ME OF PLATE MP
EDPCTH	THETA COMPONENT OF FIELD DIFFRACTED FROM CORNERS OF EDGE ME OF PLATE MP
EDPPH	PHI COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP IN RCS
EDPTH	THETA COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP IN RCS
EDRCP	PHI COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP AND REFLECTED FROM THE CYLINDER
EDRCT	THETA COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP AND REFLECTED FROM THE CYLINDER
EDRPP	PHI COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP AND THEN REFLECTED BY PLATE MR (EDGE DIF.)
EDRPT	THETA COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP AND THEN REFLECTED BY PLATE MR (EDGE DIF.)
EIPH	PHI COMPONENT OF DIRECT FIELD FROM SOURCE IN RCS
ELTH	THETA COMPONENT OF DIRECT FIELD FROM SOURCE IN RCS
EPR	PHI COMPONENT OF SCATTERED FIELD IN RCS
EPH	PHI COMPONENT OF TOTAL CALCULATED E FIELD IN PATTERN CUT COORDINATE SYSTEM
ERCAP	PHI COMPONENT OF FIELD REFLECTED FROM CYLINDER END CAP IN RCS
ERCAT	THETA COMPONENT OF FIELD REFLECTED FROM CYLINDER END CAP IN RCS
ERCPP	PHI COMPONENT OF GEOMETRICAL OPTICS FIELD REFLECTED FROM CYLINDER, AND THEN REFLECTED FROM PLATE MR
ERCPH	THETA COMPONENT OF GEOMETRICAL OPTICS FIELD REFLECTED FROM CYLINDER, AND THEN REFLECTED FROM PLATE MR
ERDPH	PHI COMPONENT OF FIELD REFLECTED FROM CYLINDER AND DIFFRACTED BY EDGE ME OF PLATE MP
ERDTH	THETA COMPONENT OF FIELD REFLECTED FROM CYLINDER AND DIFFRACTED BY EDGE ME OF PLATE MP
ERPCP	PHI COMPONENT OF GEOMETRICAL OPTICS FIELD REFLECTED BY PLATE MR AND THEN SCATTERED BY THE CYLINDER
ERPCT	THETA COMPONENT OF GEOMETRICAL OPTICS FIELD REFLECTED BY PLATE MR AND THEN SCATTERED BY THE CYLINDER
ERPDCT	PHI COMPONENT OF FIELD REFLECTED BY PLATE MR AND

DIFFRACTED BY THE CORNERS OF EDGE ME OF PLATE MP
(CORNER DIFFRACTION)

ERPDCT THETA COMPONENT OF FIELD REFLECTED BY PLATE MR AND
DIFFRACTED BY THE CORNERS OF EDGE ME OF PLATE MP
(CORNER DIFFRACTION)

ERPDUP PHI COMPONENT OF FIELD REFLECTED BY PLATE MR AND
DIFFRACTED BY EDGE ME OF PLATE MP (EDGE DIFFRACTION)

ERPDTH THETA COMPONENT OF FIELD REFLECTED BY PLATE MR AND
DIFFRACTED BY EDGE ME OF PLATE MP (EDGE DIFFRACTION)

ERPHI PHI COMPONENT OF GEOMETRICAL OPTICS FIELD
REFLECTED FROM CYLINDER

ERPPH PHI COMPONENT OF FIELD REFLECTED FROM PLATE MP IN RCS

ERPSP PHI COMPONENT OF FIELD REFLECTED BY PLATE MR
AND THEN SCATTERED BY THE CYLINDER

ERPST THETA COMPONENT OF FIELD REFLECTED BY PLATE MR,
AND THEN SCATTERED BY THE CYLINDER

ERPTH THETA COMPONENT OF FIELD REFLECTED FROM PLATE MP

ERKPP PHI COMPONENT OF FIELD REFLECTED FROM PLATE MP
AND THEN PLATE MPP IN RCS

ERKPT THETA COMPONENT OF FIELD REFLECTED FROM PLATE MP
AND THEN PLATE MPP IN RCS

ERSPP PHI COMPONENT OF FIELD SCATTERED BY THE CYLINDER
AND THEN REFLECTED BY PLATE MR

ERSPT THETA COMPONENT OF FIELD SCATTERED BY THE CYLINDER
AND THEN REFLECTED BY PLATE MR

ERTH THETA COMPONENT OF GEOMETRICAL OPTICS FIELD
REFLECTED FROM CYLINDER

ESPH PHI COMPONENT OF FIELD SCATTERED BY CYLINDER IN RCS

ESTH THETA COMPONENT OF FIELD SCATTERED BY CYLINDER IN RCS

ETH THETA COMPONENT OF SCATTERED FIELD IN RCS

ETHI THETA COMPONENT OF TOTAL CALCULATED E FIELD IN PATTERN
CUT COORDINATE SYSTEM

FN WEDGE ANGLE INDICATOR OF EDGE ME OF PLATE MP

FRQG THE FREQUENCY IN GIGAHERTZ

I DO LOOP VARIABLE

IBP PATTERN ANGLE LOWER LIMIT PLUS ONE

IEP PATTERN ANGLE UPPER LIMIT PLUS ONE

II DO LOOP VARIABLE USED TO STEP THROUGH PATTERN ANGLE

IK CHARACTER STRING USED TO INPUT COMMAND DESIRED

IS INCREMENT ON PATTERN ANGLE

IT CHARACTER STRINGS CONTAINING COMMAND VARIABLES FOR
DATA INPUT

ITI CHARACTER STRINGS USED AS COMMAND VARIABLES FOR
DATA INPUT

J DO LOOP VARIABLE USED TO STEP THROUGH INDIVIDUAL
GTD TERMS

K DO LOOP VARIABLE USED TO STEP THRU MAJOR GTD GROUPINGS

LABEL CHARACTERS USED TO SPECIFY UNITS USED TO INPUT DATA

LAMP LOGICAL VARIABLE SET TRUE IF NEC SOURCE DATA WAS
INPUT

LNROT LOGICAL VARIABLE: SET TRUE IF RCS TRANSFORMATION IS NOT
TO BE PERFORMED

MC INDEX VARIABLE FOR CORNERS

ME INDEX VARIABLE FOR EDGES

MEDA MAXIMUM NUMBER OF EDGES ALLOWED ON ONE PLATE

MEY NUMBER OF EDGES ON PLATE MP (NOT AN ARRAY)

MP INDEX VARIABLE FOR PLATES

MPDA MAXIMUM NUMBER OF PLATES ALLOWED

APP INDEX VARIABLE FOR PLATES

MR INDEX VARIABLE FOR PLATES

MS INDEX VARIABLE FOR SOURCES

MSDA MAXIMUM NUMBER OF SOURCES ALLOWED

N INDEX VARIABLE

NI INDEX VARIABLE

NJ INDEX VARIABLE

PHI PHI ANGLE DEFINING PATTERN ANGLE IN PATTERN CUT

COORDINATE SYSTEM

PHPR PHI COMPONENT OF PATTERN ANGLE IN PAT CUT COORD SYS
 PHSR PHI COMPONENT OF PATTERN (OBSERVATION) ANGLE IN RCS
 SAS SINE OF AS
 SASP $\sin(AS-PI/2)$
 SNC SINE OF THPR AND THNR
 SPs SINE OF PHSh
 STHS SINE OF THSR
 THP THETA ANGLE DEFINING PATTERN ANGLE IN PATTERN CUT
 COORDINATE SYSTEM

THPR THETA COMPONENT OF PATTERN ANGLE IN PAT CUT COORD SYS
 THSR THETA COMPONENT OF PATTERN (OBSERVATION) ANGLE IN RCS
 THNR ANGLE NEGATIVE END CAP MAKES WITH Z AXIS (IN X-Z PLANE)
 THPR ANGLE POSITIVE END CAP MAKES WITH Z AXIS (IN X-Z PLANE)
 TPPD PATTERN ANGLE WHICH REMAINS CONSTANT
 UNIT CONVERSION FACTORS TO CONVERT FROM METERS, FEET,
 OR INCHES TO METERS

VXS X,Y,Z COMPONENTS DEFINING SOURCE COORDINATE
 AXES IN RCS COMPONENTS

WI (COMPLEX) WEIGHTING COEFFICIENT OF SOURCE EXCITATION

XCL } X,Y,Z COMPONENTS DEFINING AXES OF CYLINDER
 YCL } COORDINATE SYSTEM (BEFORE RCS TRANSFORMATION)
 ZCL } (IN RCS COMPONENTS)
 XCC } X,Y,Z COMPONENTS OF LOCATION OF CYLINDER COORDINATE
 SYSTEM ORIGIN IN RCS (BEFORE RCS TRANSFORMATION)

XCGA DISTANCE BETWEEN RCS ORIGIN AND CYLINDER COORDINATE
 SYSTEM ORIGIN

XOO CONSTANT ($=0,0,0$)

XPC } X,Y,Z COMPONENTS DEFINING AXES OF PATTERN CUT COORDINATE
 YPC } SYSTEM AFTER RCS TRANSFORMATION
 ZPC } (IN RCS COMPONENTS)
 XPD } X,Y,Z COMPONENTS DEFINING AXES OF PATTERN
 YPD } CUT COORDINATE SYSTEM (IN RCS COMPONENTS)
 ZPD } (BEFORE RCS TRANSFORMATION)

XS X,Y,Z COMPONENTS OF SOURCE LOCATION (INSIDE SOURCE
 LOOP)

XXA COMPUTATIONAL VARIABLE

ZC POINT WHERE UPPER AND LOWER CYLINDER END CAPS MEET THE
 Z AXIS OF THE RCS

BABS

PURPOSE

This function computes the absolute value of a complex argument. It is similar to CABS, except it avoids run time errors when the real part and imaginary part of the argument are zero.

METHOD

The system function CABS is used unless the absolute value of the real part and the imaginary part of the argument are close to zero, in which case a very small value is returned.

SYMBOL DICTIONARY

X	ABSOLUTE VALUE OF THE REAL PART OF Z
Y	ABSOLUTE VALUE OF THE IMAGINARY PART OF Z
Z	THE COMPLEX ARGUMENT

CODE LISTING

```
1 C-----
2      FUNCTION BABS(Z)
3 C!!!
4 C!!! THIS ROUTINE IS USED TO GIVE COMPLEX ABSOLUTE VALUES. IT IS
5 C!!! USED RATHER THAN STANDARD ROUTINES TO AVOID EXECUTION
6 C!!! ERRORS.
7 C!!!
8      COMPLEX Z
9      X=ABS(REAL(Z))
10     Y=ABS(AIMAG(Z))
11     IF(X.LT.1.E-10.AND.Y.LT.1.E-10) GO TO 10
12     BABS=CABS(Z)
13     RETURN
14 10   BABS=1.E-10
15     RETURN
16     END
```

BLOCK DATA

PURPOSE

To load commonly used data into the common area.

CODE LISTING

```
1 C-----
2      BLOCK DATA
3 C!!!
4 C!!!  LOAD COMMONLY USED DATA INTO COMMON AREA.
5 C!!!
6      COMPLEX CJ,CPI4, TOP
7      COMMON/PIS/PI,TPI,DPR,RPD
8      COMMON/COMP/CJ,CPI4
9      COMMON/LOPD/TOP
10     DATA PI,TPI,DPR,RPD/3.14159265,6.28318531,57.2957795,
11     20.0174532925/
12     DATA CJ,CPI4/(0.,1.),(.70710678,-.70710678)/
13     DATA TOP/(-.70710678,.70710678)/
14     END
```

BLOG10

PURPOSE

This function computes the logarithm to the base ten of the argument. It is similar to ALOG10, except it avoids run time errors when the argument is zero.

METHOD

The system function ALOG10 is used unless the argument is close to zero, in which case the logarithm of the limit number is returned.

SYMBOL DICTIONARY

X THE ARGUMENT OF THE FUNCTION

CODE LISTING

```
1 C-----  
2      FUNCTION BLOG10(X)  
3 C!!!  
4 C!!! THIS ROUTINE AVOIDS THE ERROR ASSOCIATED WITH THE  
5 C!!! ALOG10 OF A ZERO NUMBER.  
6 C!!!  
7      IF(X.GT.1.E-10) GO TO 1  
8      BLOG10=-10.  
9      RETURN  
10 1    BLOG10=ALOG10(X)  
11      RETURN  
12      END
```

BTAN2

PURPOSE

This function computes the two argument arctangent function. It is similar to ATAN2, except it avoids run time errors when the second argument is zero.

METHOD

The system function ATAN2(Y,X) is used to return the angle in radians, whose sine is Y and cosine is X unless the second argument or both of the arguments are zero. If the second argument is zero, either $\pi/2$ or $-\pi/2$ is returned depending on the sign of the first argument. If both arguments are zero, a zero value is returned.

SYMBOL DICTIONARY

X	SECND ARGUMENT, WHICH IS THE COSINE OF THE ANGLE TO BE COMPUTED
Y	FIRST ARGUMENT, WHICH IS THE SINE OF THE ANGLE TO BE COMPUTED

CODE LISTING

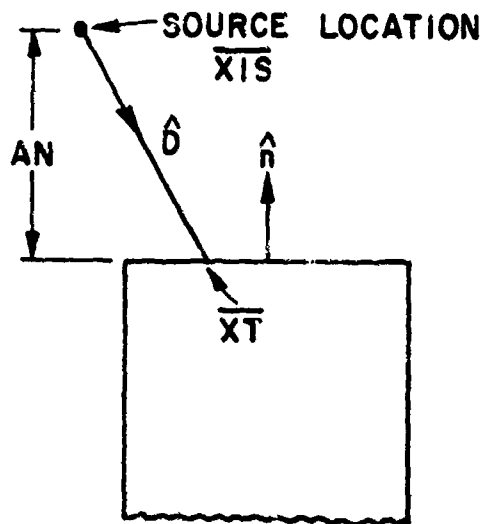
```
1 C-----
2      FUNCTION BTAN2(Y,X)
3 C!!!
4 C!!! THIS ROUTINE IS USED TO COMPUTE THE ARCTANGENT. IT IS
5 C!!! SIMILAR TO ATAN2 EXCEPT IT AVOIDS THE RUN TIME ERRORS.
6 C!!!
7      COMMON/PI5/PI,TP1,IPR,RPD
8      IF(ABS(X).GT.1.E-10) GO TO 50
9      IF(ABS(Y).GT.1.E-10) GO TO 10
10     BTAN2=0.
11     RETURN
12 10     BTAN2=PI/2.
13     IF(Y.LT.0.) BTAN2=-BTAN2
14     RETURN
15 50     BTAN2=ATAN2(Y,X)
16     RETURN
17     END
```

CAPINT

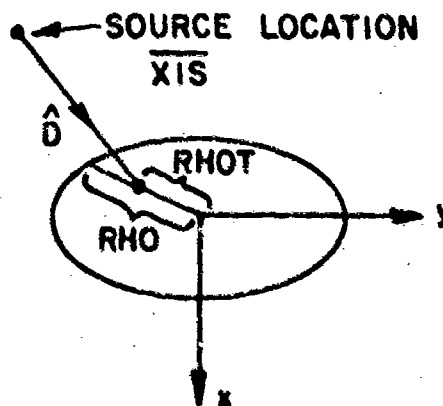
PURPOSE

To determine if a ray traveling from a given source location in a given direction will hit a cylinder end cap.

PERTINENT GEOMETRY



SIDE VIEW



TOP VIEW

Figure 48-- Geometry of ray which hits an end cap.

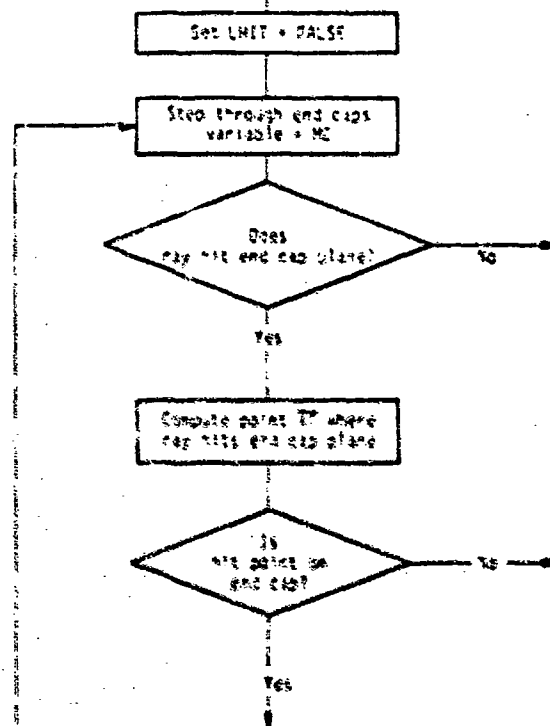
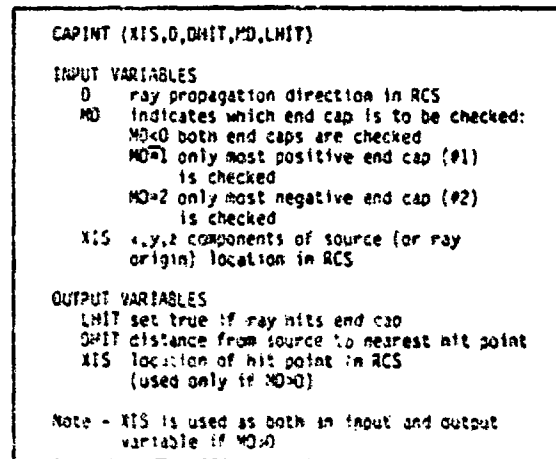
METHOD

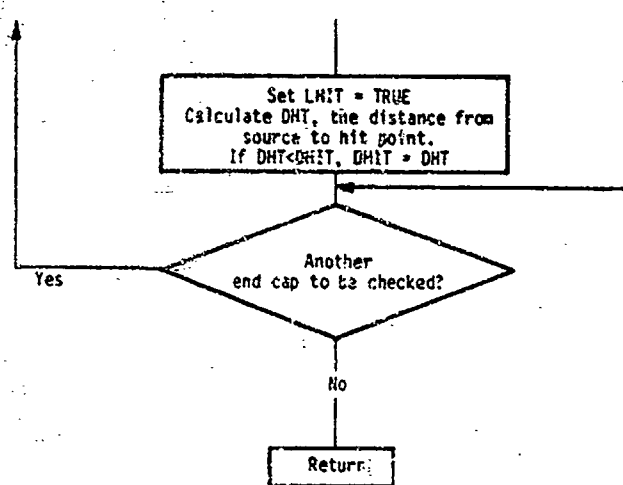
The subroutine checks to see if a ray emanating from a source in a given direction hits a cylinder end cap. First it checks if the ray is aimed toward or away from the end cap plane by comparing the sign of the dot product of the scatter direction and end cap normal (DN) and the sign of the dot product of the source location vector and end cap normal (AN). If the ray is directed toward the end cap plane as shown in Figure 48, the intersection point with the plane is found from

$$XT = XIS - \hat{D} \frac{AN}{DN}.$$

The distance from the intersection point to the center of the end cap is then compared with the radius of the end cap to determine if the intersection point lies within the finite limits of the end cap.

FLOW DIAGRAM





SYMBOL DICTIONARY

AE	DISTANCE FROM CENTER OF EDGE CAP TO EDGE ALONG LINE IN X-Z PLANE
AN	DOT PRODUCT OF VECTOR FROM END CAP TO SOURCE AND END CAP UNIT NORMAL
CVE	COSINE OF VE
D	PROPAGATION DIRECTION IN RCS
DHIT	DISTANCE FROM SOURCE TO NEAREST HIT POINT
DHT	DISTANCE FROM SOURCE TO HIT POINT
DN	DOT PRODUCT OF END CAP UNIT NORMAL AND THE RAY PROPAGATION DIRECTION
LHIT	SET TRUE IF RAY HITS END CAP
MC	END CAP INDEX VARIABLE
MD	INDICATES WHICH END CAPS ARE TO BE CHECKED
NC	SIGN CHANGE VARIABLE
RHO	DISTANCE FROM Z AXIS TO POINT WHERE RAY CONNECTING THE HIT POINT AND THE ORIGIN HITS THE CYLINDER (2-D)
RHOT	DISTANCE FROM Z AXIS TO POINT XT
SVE	SINE OF VE
VE	ELL ANGLE DEFINING HIT POINT
XIS	(ENTERING ROUTINE) SOURCE LOCATION (LEAVING ROUTINE) HIT POINT (IF MD>0)
XT	X,Y,Z COMPONENTS OF POINT WHERE RAY HITS END CAP PLANE

CODE LISTING

```

1 C-----
2 SUBROUTINE CAPINT(XIS,D,DHIT,MD,LHIT)
3 C!!!
4 C!!! DOES RAY HIT END CAP?
5 C!!!
6 DIMENSION XIS(3),D(3),XT(3)
7 LOGICAL LHIT,LDEBUG,LTEST
8 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
9 COMMON/TEST/LDEBUG,LTEST
10 LHIT=.FALSE.
11 DHIT=0.
12 C!!! STEP THRU END CAPS
13 DO 40 MC=1,2
14 IF(MD.NE.0.AND.MC.NE.MD) GO TO 40
15 NC=MC
16 IF(MC.EQ.2) NC=-1
17 AN=-XIS(1)*NC*CNC(MC)+(XIS(3)-ZC(MC))*NC*SNC(MC)
18 DN=-NC*CNC(MC)*D(1)+NC*SNC(MC)*D(3)
19 C!!! DOES RAY HIT END CAP PLANE?
20 IF(AN*DN.GE.0.) GO TO 40
21 C!!! COMPUTE POINT XT, WHERE RAY HITS END CAP PLANE
22 DO 10 N=1,3
23 XT(N)=XIS(N)-AN*D(N)/DN
24 RHOT=XT(1)*XT(1)+XT(2)*XT(2)+(XT(3)-ZC(MC))*(XT(3)-ZC(MC))
25 RHOT=SQRT(RHOT)
26 AE=A/SNC(MC)
27 C!!! IS HIT POINT ON END CAP?
28 IF(RHOT.GT.AE.AND.RHOT.GT.B) GO TO 40
29 IF(RHOT.LT.AE.AND.RHOT.LT.B) GO TO 20
30 VE=BTAN2(A*XT(2),B*XT(1))
31 CVE=COS(VE)
32 SVE=SIN(VE)
33 RHO=SQRT(AE*AE*CVE*CVE+B*B*SVE*SVE)
34 IF(RHOT.GT.RHO) GO TO 40
35 20 CONTINUE
36 C!!! CALCULATE DHT, THE DISTANCE FROM SOURCE TO HIT POINT
37 DHT=0.
38 DO 30 N=1,3
39 30 DHT=DHT+(XT(N)-XIS(N))*(XT(N)-XIS(N))
40 DHT=SQRT(DHT)+1.E-5
41 IF(LHIT.AND.(DHT.GT.DHIT)) GO TO 40
42 LHIT=.TRUE.
43 DHIT=DHT
44 IF(MD.LE.0) GO TO 40
45 DO 35 N=1,3
46 35 XIS(N)=X1(N)
47 40 CONTINUE
48 IF(.NOT.LTEST) RETURN
49 WRITE(6,900)
50 900 FORMAT(/,' TESTING CAPINT SUBROUTINE')
51 WRITE(6,*) XIS
52 WRITE(6,*) D
53 WRITE(6,*) DHIT,MD,LHIT
54 RETURN
55 END

```

CYLINT

PURPOSE

To determine if a ray travelling from a given source location in a given direction will intersect the elliptic cylinder.

PERTINENT GEOMETRY

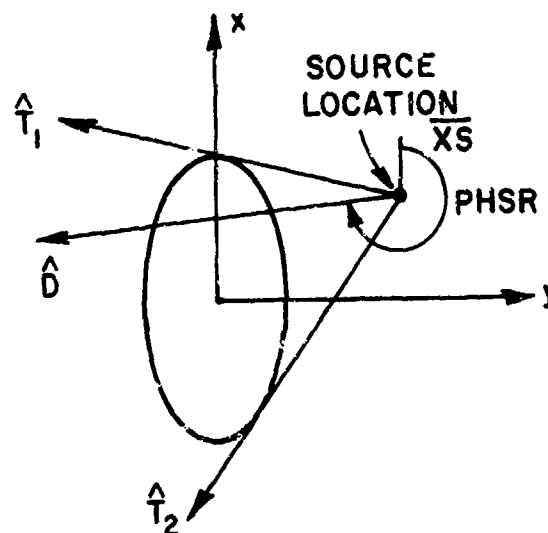


Figure 49a--Illustration of ray that hits infinite cylinder.

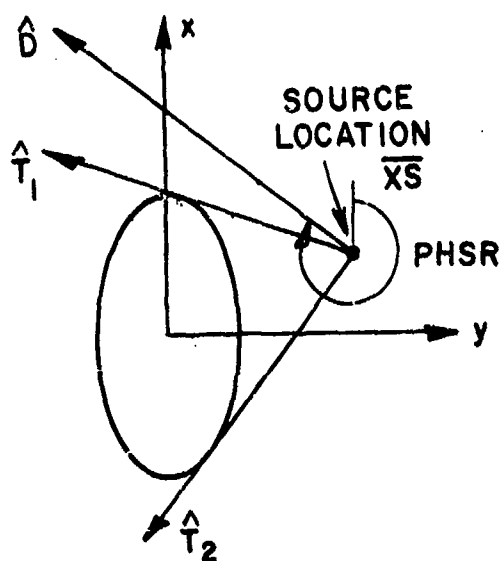


Figure 49b--Illustration of ray that doesn't hit finite cylinder.

$$\hat{T}_1 = \hat{x} \text{ BT}(1) + \hat{y} \text{ BT}(2)$$

$$\hat{T}_2 = \hat{x} \text{ BT}(3) + \hat{y} \text{ BT}(4)$$

$$\hat{D} = \hat{x} \text{ D}(1) + \hat{y} \text{ D}(2) + \hat{z} \text{ D}(3)$$

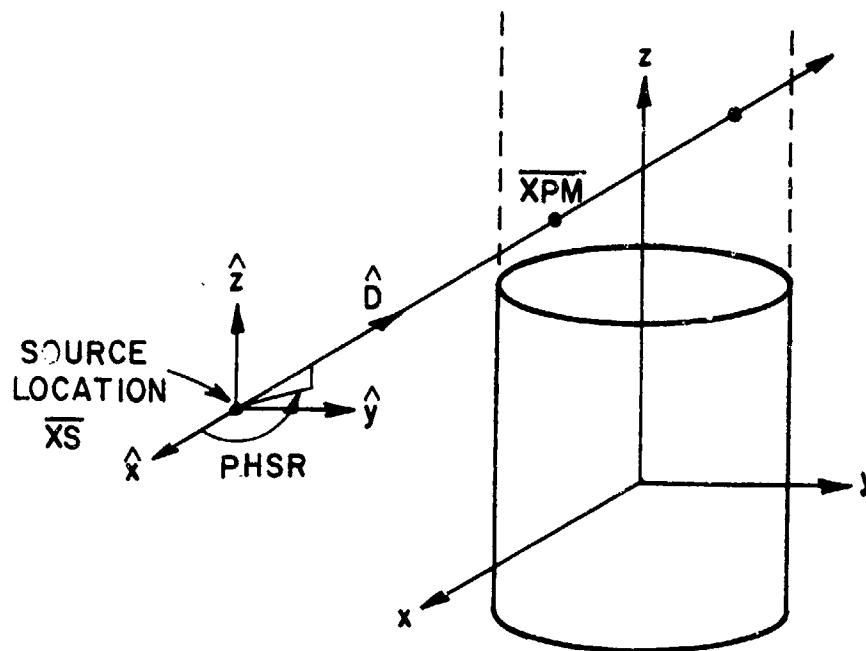


Figure 50a--Illustration of ray that hits infinite cylinder but not finite cylinder.

$$\overline{XPM} = \hat{x} \text{ XPM}(1) + \hat{y} \text{ XPM}(2) + \hat{z} \text{ XPM}(3)$$

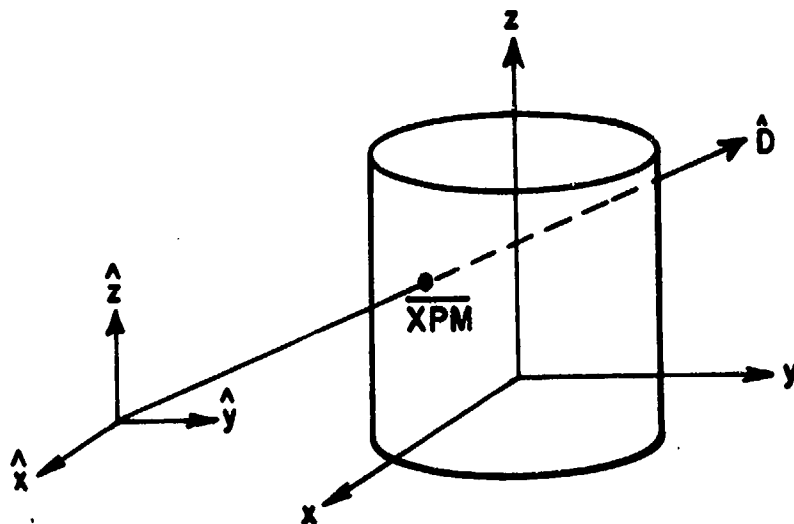


Figure 50b--Illustration of ray that hits finite cylinder.

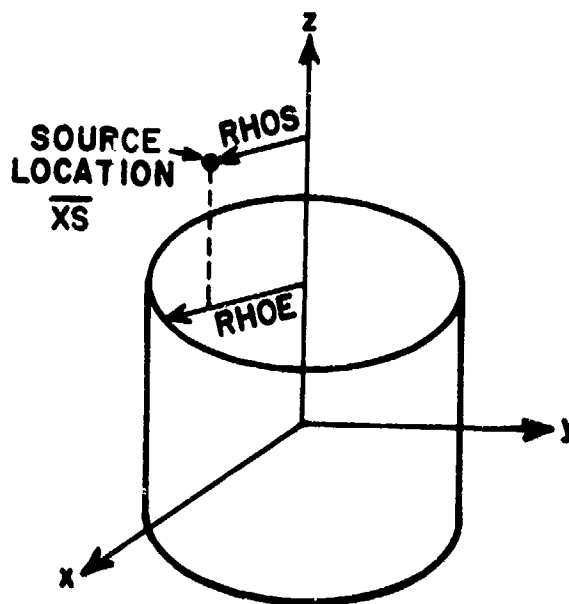


Figure 51--Illustration of source which cannot illuminate curved cylinder surface. $RHOS < RHOE$.

METHOD

This subroutine determines if a ray emanating from a source in a given direction hits the finite elliptic cylinder. First the distance from the source to the cylinder axis is compared to the radius of the cylinder to see if the source can illuminate the curved surface of the cylinder as illustrated in Figure 51. If it can not, then the subroutine checks whether the ray hits an end cap. If it is possible to hit the curved surface, the ray is checked to see whether or not it is aimed in the direction of the infinite cylinder as shown in Figure 49.

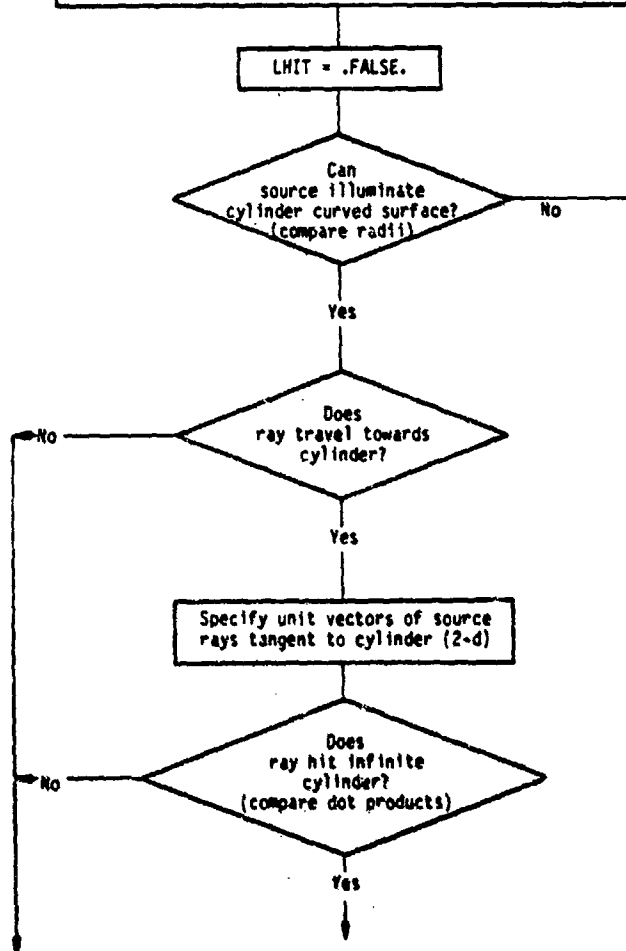
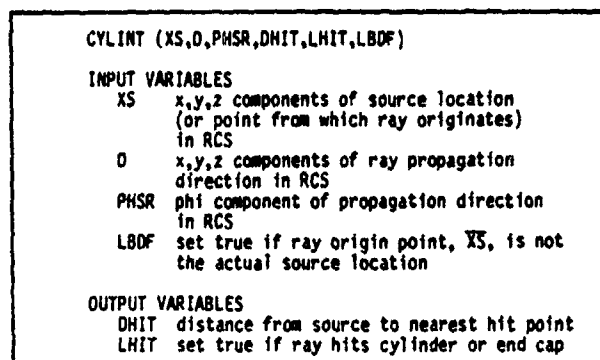
If the ray travels towards the cylinder, the routine compares dot products in order to determine if the ray will hit the infinite cylinder:

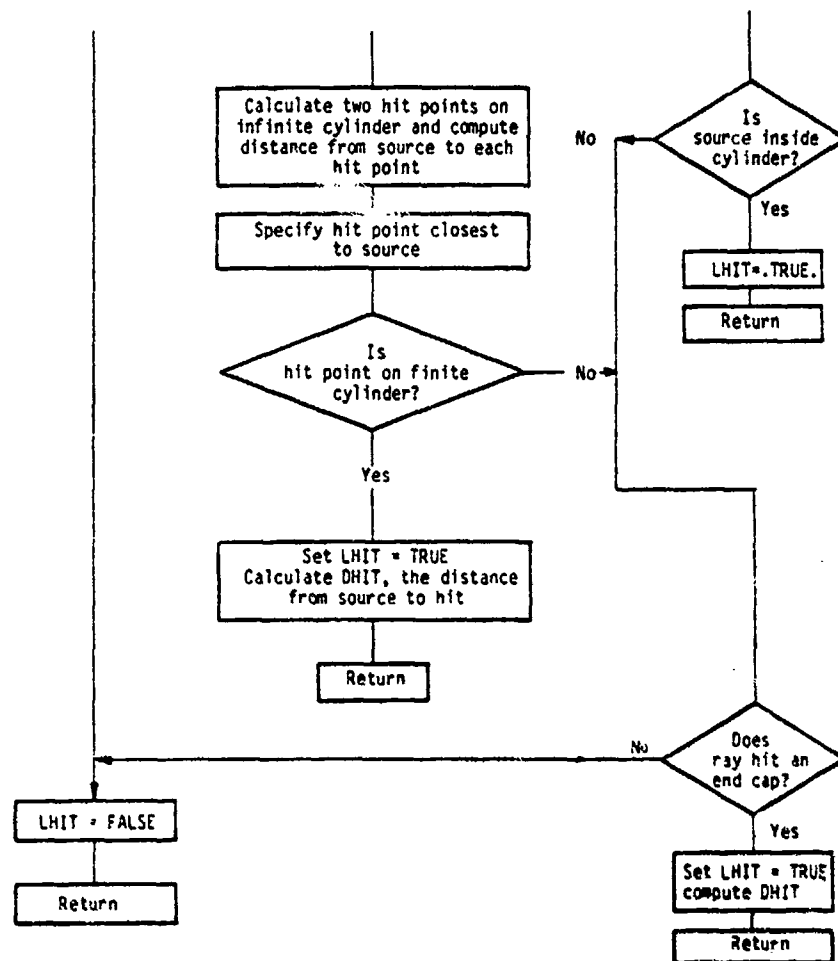
If $\hat{D} \cdot \hat{T}_1 \geq \hat{T}_1 \cdot \hat{T}_2$ and $\hat{D} \cdot \hat{T}_2 \geq \hat{T}_1 \cdot \hat{T}_2$, the ray hits the infinite cylinder (see Figure 49a).

If $\hat{D} \cdot \hat{T}_1 < \hat{T}_1 \cdot \hat{T}_2$ or $\hat{D} \cdot \hat{T}_2 < \hat{T}_1 \cdot \hat{T}_2$, the ray does not hit the infinite cylinder (see Figure 49b).

The subroutine then solves a quadratic equation to determine the intersection point. The details are given on pages 90-96 of Reference 1. A test is then made as to whether or not this intersection point lies on or off the limits of the finite cylinder (see Figures 50a and 50b).

FLOW DIAGRAM





SYMBOL DICTIONARY

BM	PARAMETER USED IN COMPUTING HIT POINT 1
BPL	PARAMETER USED IN COMPUTING HIT POINT 2
BTD	X AND Y COMPONENTS OF UNIT VECTORS OF SOURCE RAYS TANGENT TO CYLINDER
CPS	COSINE OF PHSR
CVE	COSINE OF VE
D	RAY PROPAGATION DIRECTION IN REF COORD SYS
D12	DOT PRODUCT OF SOURCE VECTORS TANGENT TO THE CYLINDER (IN X-Y PLANE)
DD1	DOT PRODUCT OF THE PROPAGATION DIRECTION AND T1 TANGENT UNIT VECTOR
DD2	DOT PRODUCT OF THE PROPAGATION DIRECTION AND T2 TANGENT UNIT VECTOR
DHIT	DISTANCE FROM SOURCE TO (NEAREST) HIT POINT
DM	DISTANCE FROM SOURCE TO HIT POINT 1
DPL	DISTANCE FROM SOURCE TO HIT POINT 2
DTD	DOT PRODUCT OF SOURCE VECTORS TANGENT TO THE CYLINDER (X-Y PLANE)
DXY	DOT PRODUCT OF RAY FROM ORIGIN TO SOURCE AND PROPAGATION DIRECTION (IN X-Y PLANE)
F	COMPUTATIONAL VARIABLE
FG	COMPUTATIONAL VARIABLE
FGH	COMPUTATIONAL VARIABLE
FH	COMPUTATIONAL VARIABLE
G	COMPUTATIONAL VARIABLE
GH	COMPUTATIONAL VARIABLE
H	COMPUTATIONAL VARIABLE
LBDF	SET TRUE IF RAY ORIGIN XS IS NOT THE SOURCE LOCATION
LHIT	SET TRUE IF RAY HITS CYLINDER OR END CAP
PHSR	PHI COMPONENT OF PROPAGATION DIRECTION IN RCS
RHOE	RADIUS FROM Z AXIS TO POINT WHERE RAY FROM ORIGIN TO SOURCE INTERSECTS THE CYLINDER
RHOS	DISTANCE FROM SOURCE TO Z AXIS
SPS	SINE OF PHSR
SVE	SINE OF VE
TOP	COMPUTATIONAL VARIABLE
TX1	X COMPONENT OF TANGENT UNIT VECTOR, T1
TX2	X COMPONENT OF TANGENT UNIT VECTOR, T2
TY1	Y COMPONENT OF TANGENT UNIT VECTOR, T1
TY2	Y COMPONENT OF TANGENT UNIT VECTOR, T2
VE	ELL ANGLE OF SOURCE LOCATION IN ERCS
VM	ELL ANGLE DEFINING HIT POINT 1 ON CYLINDER IN ERCS
VPL	ELL ANGLE DEFINING HIT POINT 2 ON CYLINDER IN ERCS
VT	ELL ANGLE DEFINING HIT POINT ON CYLINDER CLOSEST TO SOURCE
VTD	NOT USED
XPM	{ USED IN SEVERAL CASES TO DEFINE HIT POINT (X,Y,Z COMPONENTS IN RCS) ON CYLINDER (USED IN VARIOUS FORMS)
YPM	
ZPM	
XS	SOURCE LOCATION (OR POINT FROM WHICH RAY ORIGINATES) IN RCS

CODE LISTING

```

1 C-----
2 SUBROUTINE CYLINT(XS,D,PHSR,DHIT,LHIT,LBDF)
3 C!!!
4 C!!! DOES RAY HIT CYLINDER?
5 C!!!
6 DIMENSION D(3),XS(3),VTD(2),BTD(4)
7 LOGICAL LHIT,LBDF,LPLA,LCYL,LDEBUG,I TEST
8 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CIC(2)
9 COMMON/PI5/PI,TPI,DPR,RPD
10 COMMON/ENDSCL/DTS,VTS(2),BTS(4)
11 COMMON/LPLCY/LPLA,LCYL
12 COMMON/TEST/LDEBUG,LTEST
13 LHIT=.FALSE.
14 DHIT=0.
15 IF(.NOT.LCYL) GO TO 50
16 RHOS=SQRT(XS(1)*XS(1)+XS(2)*XS(2))
17 C!!! CAN SOURCE ILLUMINATE CYLINDER SURFACE?
18 IF(RHOS.GT.A.AND.RHOS.GT.B) GO TO 5
19 IF(RHOS.LT.A.AND.RHOS.LT.B) GO TO 30
20 VE=ATAN2(A*XS(2),B*XS(1))
21 CVE=COS(VE)
22 SVE=SIN(VE)
23 RHOE=SQRT(A*A*CVE*CVE+B*B*SVE*SVE)
24 IF(RHOS.LE.RHOE) GO TO 30
25 5 CONTINUE
26 CPS=COS(PHSR)
27 SPS=SIN(PHSR)
28 DXY=XS(1)*CPS+XS(2)*SPS
29 C!!! DOES RAY TRAVEL TOWARDS CYLINDER?
30 C!!! (CHECK SIGN OF DOT PRODUCT OF PROP. DIR AND
31 C!!! SOURCE LOCATION VECTOR)
32 IF(DXY.GT.0.) GO TO 50
33 IF(LJDF) GO TO 10
34 C!!! SPECIFY CYLINDER TANGENT UNIT VECTORS
35 D12=DTS
36 TX1=BTS(1)
37 TY1=BTS(2)
38 TX2=BTS(3)
39 TY2=BTS(4)
40 GO TO 20
41 10 CALL TANG(BTD,VTD,BTD,XS)
42 D12=DTD
43 TX1=BTD(1)
44 TY1=BTD(2)
45 TX2=BTD(3)
46 TY2=BTD(4)
47 20 CONTINUE
48 DD1=CPS*TX1+SPS*TY1
49 DD2=CPS*TX2+SPS*TY2
50 C!!! COMPARE DOT PRODUCTS TO DETERMINE IF RAY HITS
51 C!!! INFINITE CYLINDER
52 IF(DD1.LT.D12.OR.DD2.LT.D12) GO TO 50
53 F=A*SPS
54 G=B*CPX
55 H=XS(1)*SPS-XS(2)*CPS
56 PH=PI/2
57 FG=F*G*G
58 PH=PI-PI/2
59 FGH=PH*FG*G
60 IF(FGH.LT.0.) GO TO 50
61 DPL=(F*G*G+SQRT(FGH))/FG
62 DM=(F*G*G-SQRT(FGH))/FG
63 TOP=(1.-DM*PL)/A
64 VPL=BLA2(TOP,BPL)
65 TOP=(1.-DM*PL)/B
66 C!!! COMPUTE TWO HIT POINTS AND COMPUTE DISTANCE

```

```

07 C!!! FROM SOURCE TO EACH POINT
08 VM=BTAN2(TOP,BM)
09 XPM=A*COS(VPL)
10 YPM=B*SIN(VPL)
11 DPL=SQRT((XPM-XS(1))**2+(YPM-XS(2))**2)
12 XPM=A*COS(VM)
13 YPM=B*SIN(VM)
14 DM=SQRT((XPM-XS(1))**2+(YPM-XS(2))**2)
15 C!!! SPECIFY HIT POINT CLOSEST TO SOURCE
16 VT=VM
17 IF(DPL.LE.DM) VT=VPL
18 XPM=A*CCS(VT)
19 ZPM=D(3)*(XPM-XS(1))/D(1)
20 ZPS=ZPM+XS(3)
21 C!!! IS HIT POINT ON FINITE CYLINDER?
22 IF(ZPS.GT.ZC(1)+XPM*CTC(1).OR.
23 2ZPS.LT.ZC(2)+XPM*CTC(2)) GO TO 40
24 XPM=XPM-XS(1)
25 YPM=B*SIN(VT)-XS(2)
26 C!!! CALCULATE DISTANCE FROM SOURCE TO HIT
27 DHIT=SQRT(XPM*XPM+YPM*YPM+ZPM*ZPM)*.E-5
28 LHIT=.TRUE.
29 GO TO 50
30 CONTINUE
31 C!!! IF SOURCE CANNOT ILLUMINATE CYLINDER SIDES, IS SOURCE
32 C!!! INSIDE CYLINDER?
33 IF(XS(3).GT.(ZC(1)+XS(1)*CTC(1))) GO TO 40
34 IF(XS(3).LT.(ZC(2)+XS(1)*CTC(2))) GO TO 40
35 LHIT=.TRUE.
36 GO TO 50
37 40 CONTINUE
38 C!!! IF RAY IS NOT SHADOWED BY CYLINDER, CHECK TO SEE IF RAY
39 C!!! HITS END CAP
40 CALL CAPINT(XS,D,DHIT,0,LHIT)
41 50 IF(.NOT.LTEST) RETURN
42 WRITE(6,900)
43 600 FORMAT(/,' TESTING CYLINT SUBROUTINE')
44 WRITE(6,*) XS
45 WRITE(6,*) D
46 WRITE(6,*) PMSH,DHIT,LHIT,LDDF
47 RETURN
48 END

```

DFPTCL

PURPOSE

To determine the four diffraction points which can occur on a cylinder end cap rim for a given radiation direction $\hat{D}$.

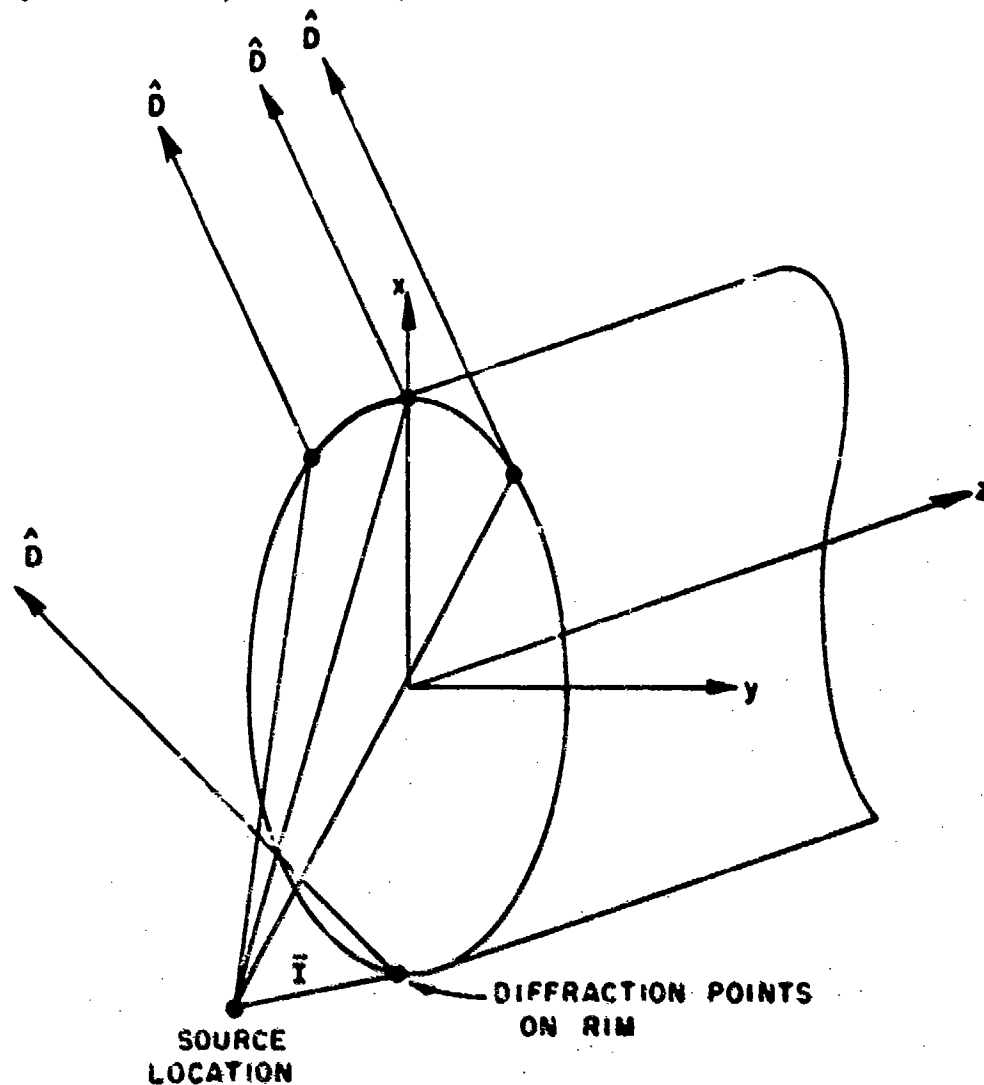


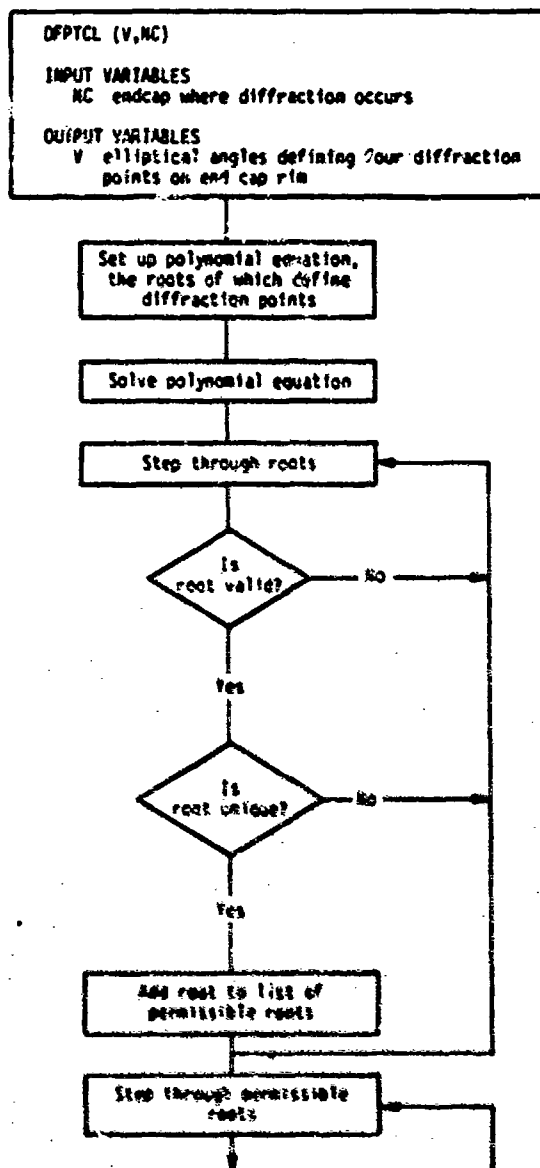
Figure 52-- Curved wedge diffraction points on rim of end cap of finite elliptic cylinder.

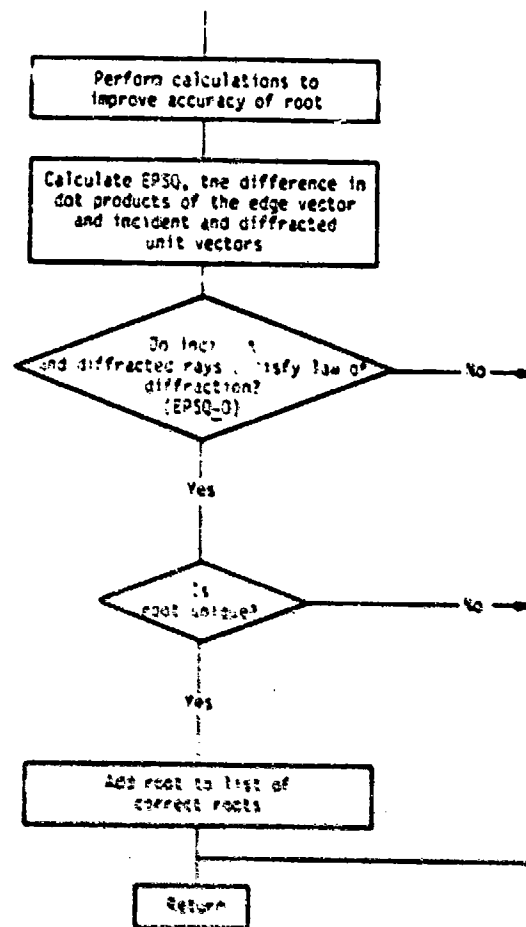
METHOD

An eighth order polynomial equation is used to solve for eight possible points on the end cap rim that can be diffraction points. These points are defined by elliptic angles in the local elliptic coordinate system for the end cap. The points are next integerized and sorted to remove duplicate points. The accuracy of the possible

diffraction points are then improved by a first order Taylor series interpolation scheme. The details are given on pages 125-127 of Reference 1. The two to four correct diffraction points are verified by checking to see which of the remaining points satisfy the laws of diffraction.

FLOW DIAGRAM





SYMBOL DICTIONARY

AE	HALF LENGTH OF END CAP (HALF LENGTH OF LINE CREATED BY INTERSECTION OF END CAP AND XZ PLANE)
C	COSINE OF VR
CC	POLYNOMIAL EQ. COEFFICIENTS
CV	COMPUTATIONAL VARIABLE
D4	COMPUTATIONAL VARIABLE
DD	COMPUTATIONAL VARIABLE
DEEX	X,Y,Z COMPONENTS OF VECTOR FROM DIFFRACTION
DEEY	
DEEZ	
DEL	POINT TO CENTER OF END CAP IN RCS
DEL	TEST VARIABLE
DEN1	MAGNITUDE OF UNNORMALIZED EDGE UNIT VECTOR
DEN2	DISTANCE FROM SOURCE TO IMPROVED DIFFRACTION POINT
DEN3	LENGTH OF INCIDENT RAY VECTOR
DEN5	COMPUTATIONAL VARIABLE
DM	X,Y,Z COMPONENTS OF UNIT VECTOR OF PROPAGATION DIRECTION IN END CAP COORDINATE SYSTEM
DOTQ1	DOT PRODUCT OF EDGE VECTOR AND INCIDENT RAY
DOTQ2	DOT PRODUCT OF EDGE VECTOR AND DIFFRACTED RAY
DSSX	X,Y,Z COMPONENTS OF VECTOR TANGENT TO DIFFRACTION POINT
DSSY	
DSSZ	
DV	IN END CAP PLANE IN RCS
EE	CHANGE IN ELL ANGLE V CALCULATED TO IMPROVE ACCURACY OF DIFFRACTION POINT
EEX	X,Y,Z COMPONENTS OF RAY TANGENT TO DIFFRACTION
EEY	
EEZ	
EPSO	POINT IN RCS
ERCS	DIFFERENCE IN DOTQ1 AND DOTQ2 (ERROR TEST VARIABLE)
ERCS	(NOR A VARIABLE) ABBR. FOR ELLIPTICAL REFERENCE COORDINATE SYSTEM
EX	X,Y,Z COMPONENTS OF NORMALIZED EDGE UNIT VECTOR
EY	
EZ	
I	IN RCS
IDEL	DO LOOP VARIABLE
IV	TEST VARIABLE
J	ELL ANGLES DEFINING PERMISSABLE DIFFRACTION POINTS IN ERCS (IN DEGREES, ROUNDED OFF TO NEAREST INTEGER)
K	ELL ANGLE DEFINING DIFFRACTION POINT IN ERCS IN DEG.
N	DO LOOP VARIABLE
NC	INDEX VARIABLE (ALSO NUMBER OF PERMISSABLE ROOTS)
NCC	END CAP WHERE DIFFRACTION OCCURS
P	SIGN CHANGE VARIABLE
Q	POLYNOMIAL EQ. VARIABLE
QC	POLYNOMIAL EQ. VARIABLE
R	COMPLEX CONJ. OF Q
RC	POLYNOMIAL EQ. VARIABLE
ROOT	COMPLEX CONJ. OF R
S	ROOTS OF POLYNOMIAL EQ RETURNED FROM SUB. POLYRT
SSX	SINE OF ELL ANGLE V (ALSO POLY. EQ. VARIABLE)
SSY	X,Y,Z COMPONENTS OF VECTOR INCIDENT ON EDGE
SSZ	
SXQ	
SYQ	X,Y,Z COMPONENTS OF UNIT VECTOR OF PROPAGATION DIRECTION OF INCIDENT RAY IN RCS
SZQ	
V	
VQ	ELL ANGLES DEFINING DIFFRACTION POINTS IN ERCS
VR	ELL ANGLE DEFINING DIFFRACTION POINT (IMPROVED ACCURACY)
VT	ELL ANGLE DEFINING DIFFRACTION POINT
XSM	ELL ANGLE DEFINING DIFFRACTION POINT (IMPROVED ACCURACY) IN DEGREES
YSM	X,Y,Z COMPONENTS OF SOURCE LOCATION IN
ZSM	
	END CAP COORDINATE SYSTEM

CODE LISTING

```

1 C-----
2      SUBROUTINE DFPTCL(V,NC)
3 C!!!
4 C!!! DETERMINES THE DIFFRACTION POINT ON THE CURVED
5 C!!! EDGE OF THE ELLIPTIC CYLINDER END CAP
6 C!!!
7      COMPLEX CC(9),ROOT(8),CV,0,0C,R,RC
8      DIMENSION IV(8),V(4),DV(3)
9      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
10     COMMON/SORINF/XS(3),VXS(3,3)
11     COMMON/PIS/PI,TPI,DPR,RPD
12     COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
13     NCC=NC
14     IF(NC.GT.1) NCC=-1
15     DO 10 I=1,8
16     IV(I)=-1000
17     IF(I.LE.4) V(I)=-1000.
18 10    CONTINUE
19     DM(1)=SNC(NC)*D(1)+CNC(NC)*D(3)
20     DM(2)=D(2)
21     DM(3)=-CNC(NC)*D(1)+SNC(NC)*D(3)
22     XSM=SNC(NC)*XS(1)+CNC(NC)*(XS(3)-ZC(NC))
23     YSM=XS(2)
24     ZSM=-CNC(NC)*XS(1)+SNC(NC)*(XS(3)-ZC(NC))
25     AE=A/SNC(NC)
26 C!!! SET UP POLYNOMIAL EQUATION
27     P=AE*AE-B*B
28     IF(ABS(AE-B).LT.1.E-9) P=0.
29     Q=CMPLX(AE*XSM,-B*YSM)
30     QC=CONJG(Q)
31     R=CMPLX(B*DM(2),AE*DM(1))
32     RC=CONJG(R)
33     S=AE*AE+B*B+2.*(XSM*XSM+YSM*YSM+ZSM*ZSM)
34     CC(9)=P*(P+R*R)
35     CC(8)=-4.*Q*(P+R*R)
36     CC(7)=2.*(2.*Q*Q+S*R*R+P*R*RC)
37     CC(6)=4.*(QC*(P-R*R)-2.*Q*R*RC)
38     CC(5)=CMPLX(Q,0.)
39     CC(5)=CC(5)+P*(R*R+RC*RC)-2.*(P*P+4.*Q*QC)+4.*S*R*RC
40     CC(4)=CONJG(CC(6))
41     CC(3)=CONJG(CC(7))
42     CC(2)=CONJG(CC(8))
43     CC(1)=CONJG(CC(9))
44 C!!! SOLVE POLYNOMIAL EQUATION
45     CALL POLYRT(6,CC,ROOT)
46     N=0
47 C!!! STEP THRU ROOTS
48     DO 200 I=1,8
49 C!!! CHECK TO SEE IF ROOT IS VALID
50     RM=ABS(ROOT(I))
51     IF(RM.LT.0.1) GO TO 200
52     CV=DPR*CMPLX(0.,-1.)*CLOG(ROOT(I))
53     VT=ABS(1.-RM)
54     IF(VT.GT.0.1) GO TO 200
55     IF(REAL(CV).GE.0.) J=REAL(CV)+.5
56     IF(REAL(CV).LT.0.) J=REAL(CV)-.5
57     IF(J.LT.0) J=J+360
58     IF(J.GE.360) J=J-360
59     IF(N.EQ.0) GO TO 151
60     DO 150 K=1,N
61     IDEL=IAES(J-IV(K))
62 C!!! IS ROOT UNIQUE? IF SO ADD TO LIST OF PERMISSABLE ROOTS
63     IF(IDEL.LE.1.OR.IDEL.GE.359) GO TO 200
64 150    CONTINUE
65 151    N=N+1
66     IV(N)=J

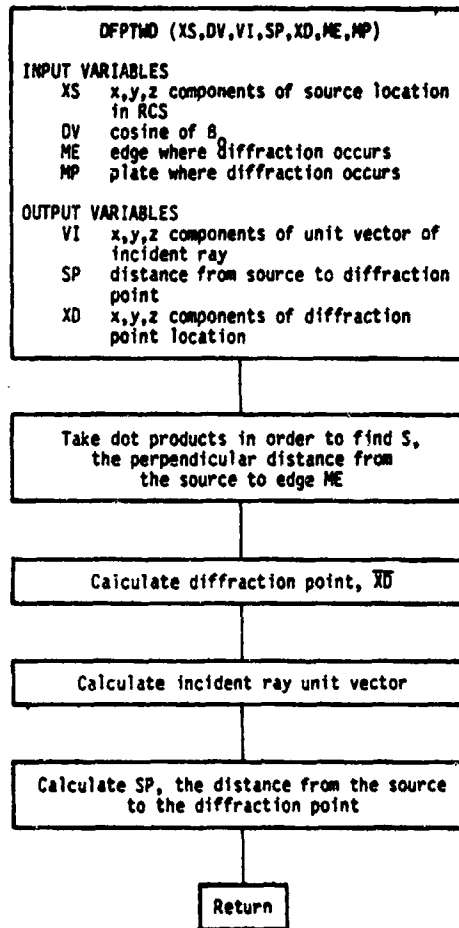
```

```

67 200 CONTINUE
68 IF(N.EQ.0) GO TO 3031
69 J=0
70 C!!! STEP THRU PERMISSABLE ROOTS
71 DO 300 I=1,N
72 C!!! PERFORM CALCULATION TO IMPROVE ACCURACY OF ROOT
73 VR=IV(I)*RPD
74 S=SIN(VR)
75 C=COS(VR)
76 DSSX=-A*S
77 DSSY=B*C
78 DSSZ=-A*CTC(NC)*S
79 DEEX=-A*C
80 DEEY=-B*S
81 DEEZ=-A*CTC(NC)*C
82 SSX=A*C-XSM
83 SSY=B*S-YSM
84 SSZ=A*CTC(NC)*C-XS(3)+ZC(NC)
85 DEN3=SQRT(SSX*SSX+SSY*SSY+SSZ*SSZ)
86 EEX=DSSX
87 EEY=DSSY
88 EEZ=DSSZ
89 DD=(EEX*SSX+EEY*SSY+EEZ*SSZ)/DEN3
90 DEN5=DEN3*(EEX*DM(1)+EEY*DM(2)+EEZ*DM(3))-EEX*SSX-EEY*SSY-EEZ*SSZ
91 D4=EEX*DSSX+EEY*DSSY+EEZ*DSSZ+DEEX*SSX+DEEY*SSY+DEEZ*SSZ
92 D4=D4-DD*(EEX*DM(1)+EEY*DM(2)+EEZ*DM(3))
93 D4=D4-DEN3*(DEEX*DM(1)+DEEY*DM(2)+DEEZ*DM(3))
94 DV=DEN5*DPR/D4
95 IF(ABS(DV).GT.2.) GO TO 300
96 VT=IV(I)+DV
97 VQ=VT*RPD
98 S=SIN(VQ)
99 C=COS(VQ)
100 DEN1=A*A*S*S+B*B*C*C+A*A*S*S*CTC(NC)*CTC(NC)
101 DEN1=SQRT(DEN1)
102 DEN2=(A*C-XSM)*(A*C-XSM)+(B*S-YSM)*(B*S-YSM)
103 DEN2=SQRT(DEN2+(A*CTC(NC)*C-XS(3)+ZC(NC))
104 2*(A*CTC(NC)*C-XS(3)+ZC(NC)))
105 EXQ=-A*S/DEN1
106 EYQ=B*C/DEN1
107 EZQ=-A*CTC(NC)*S/DEN1
108 SXQ=(A*C-XSM)/DEN2
109 SYQ=(B*S-YSM)/DEN2
110 SZQ=(A*CTC(NC)*C-XS(3)+ZC(NC))/DEN2
111 C!!! CALCULATE EPSQ, THE DIFFERENCE IN DOT PRODUCTS OF THE EDGE
112 C!!! VECTOR AND INC. AND DIF. PROPAGATION UNIT VECTORS
113 DOTQ1=SXQ*EXQ+SYQ*EYQ+SZQ*EZQ
114 DOTQ2=DM(1)*EXQ+DM(2)*EYQ+DM(3)*EZQ
115 EPSQ=DOTQ1-DOTQ2
116 C!!! DO INC. AND DIF. RAYS SATISFY LAW OF DIFFRACTION (EPSQ=0)
117 IF(ABS(EPSQ).GT.1.E-3) GO TO 300
118 IF(VT.GE.360.) VT=VT-360.
119 IF(VT.LT.0.) VT=360.+VT
120 IF(J.EQ.0) GO TO 289
121 DO 288 K=1,J
122 DEL=ABS(VT-V(K))
123 C!!! IS THE ROOT UNIQUE? IF SO, ADD TO LIST OF CORRECT ROOTS
124 IF(DEL.LT.0.5.OR.DEL.GT.359.5) GO TO 300
125 288 CONTINUE
126 289 J=J+1
127 V(J)=VT
128 300 CONTINUE
129 3031 RETURN
130 END

```


FLOW DIAGRAM



SYMBOL DICTIONARY

CTB	COTANGENT OF BETA
DV	COSINE OF BETA
ME	EDGE WHERE DIFFRACTION OCCURS
MP	PLATE WHERE DIFFRACTION OCCURS
N	DO LOOP VARIABLE
P	DOT PRODUCT OF EDGE VECTOR AND VECTOR FROM CORNER ME TO SOURCE
S	PERPENDICULAR DISTANCE FROM SOURCE TO EDGE ME
SP	DISTANCE FROM SOURCE TO DIFFRACTION POINT
SX	VARIABLE USED TO CALCULATE S
VI	INCIDENT RAY UNIT VECTOR
XD	LOCATION OF DIFFRACTION POINT
XS	SOURCE LOCATION

CODE LISTING

```

1 C-----
2 SUBROUTINE DFPTWD(XS,DV,VI,SP,XD,ME,MP)
3 C!!!
4 C!!! DETERMINATION OF THE DIFFRACTION POINT
5 C!!!
6 DIMENSION XS(3),XD(3),VI(3)
7 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
8 2,MEP(14),MPX
9 CTB=DV/SQRT(1.-DV*DV)
10 P=0.
11 DO 10 N=1,3
12 10 P=P+(XS(N)-X(MP,ME,N))*V(MP,ME,N)
13 S=0.
14 DO 20 N=1,3
15 SX=XS(N)-X(MP,ME,N)-P*V(MP,ME,N)
16 20 S=S+SX*SX
17 S=SQRT(S)
18 DO 30 N=1,3
19 30 XD(N)=X(MP,ME,N)+(S*CTB+P)*V(MP,ME,N)
20 SP=0.
21 DO 40 N=1,3
22 VI(N)=XD(N)-XS(N)
23 40 SP=SP+VI(N)*VI(N)
24 SP=SQRT(SP)
25 DO 50 N=1,3
26 50 VI(N)=VI(N)/SP
27 RETURN
28 END

```

DFRFPT

PURPOSE

To determine the ray path for a source ray which is diffracted off of a given edge on a given plate and then reflected in a given direction by the cylinder.

PERTINENT GEOMETRY

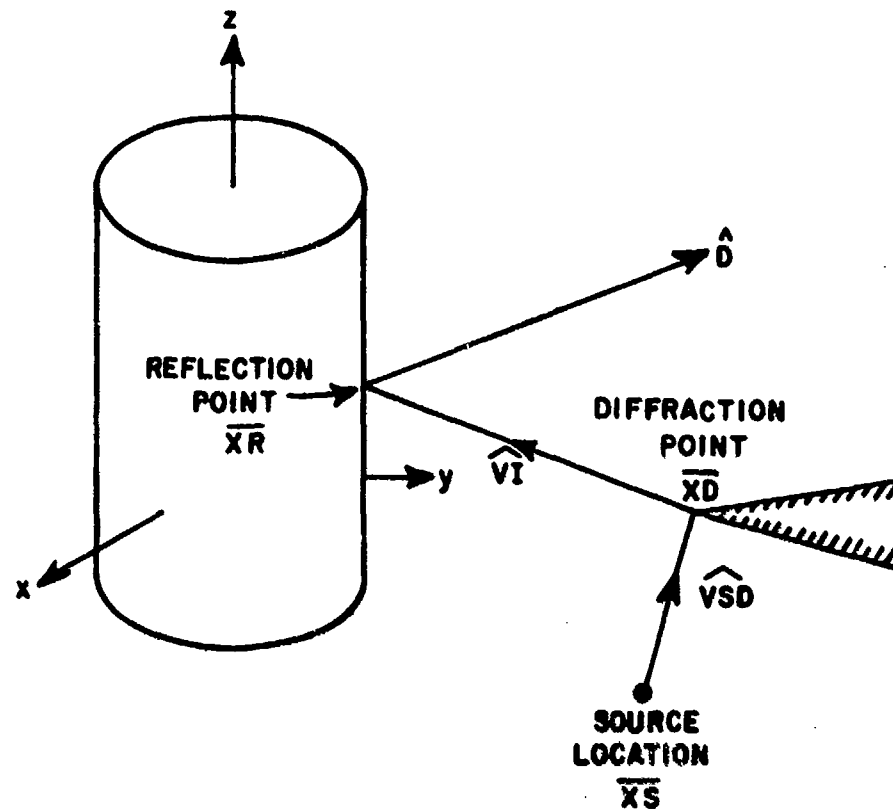
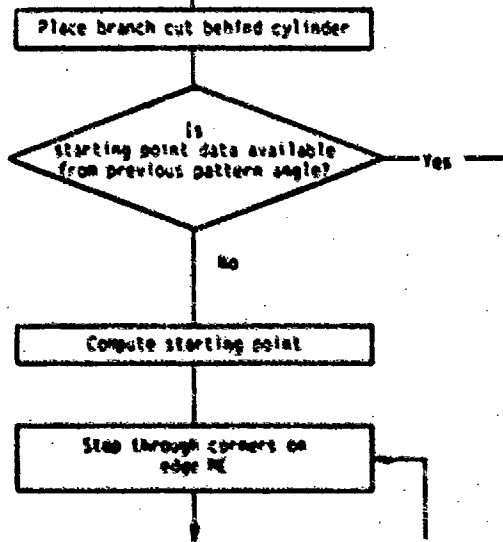
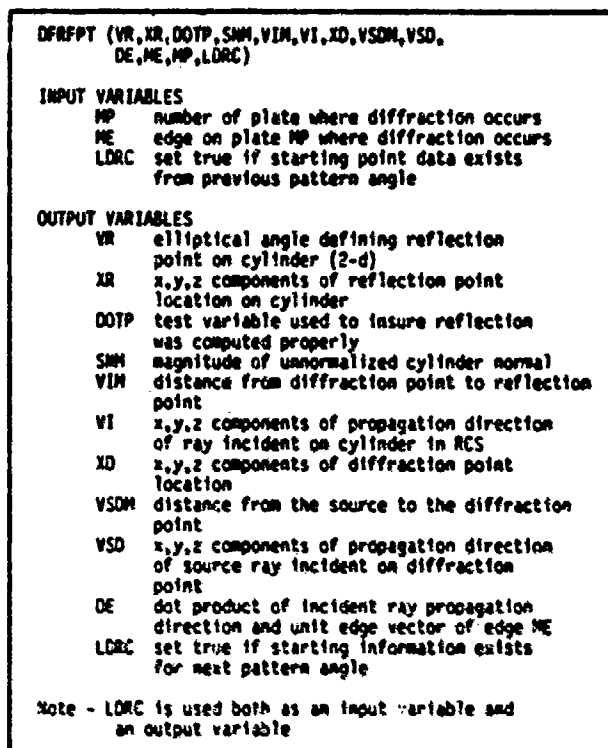


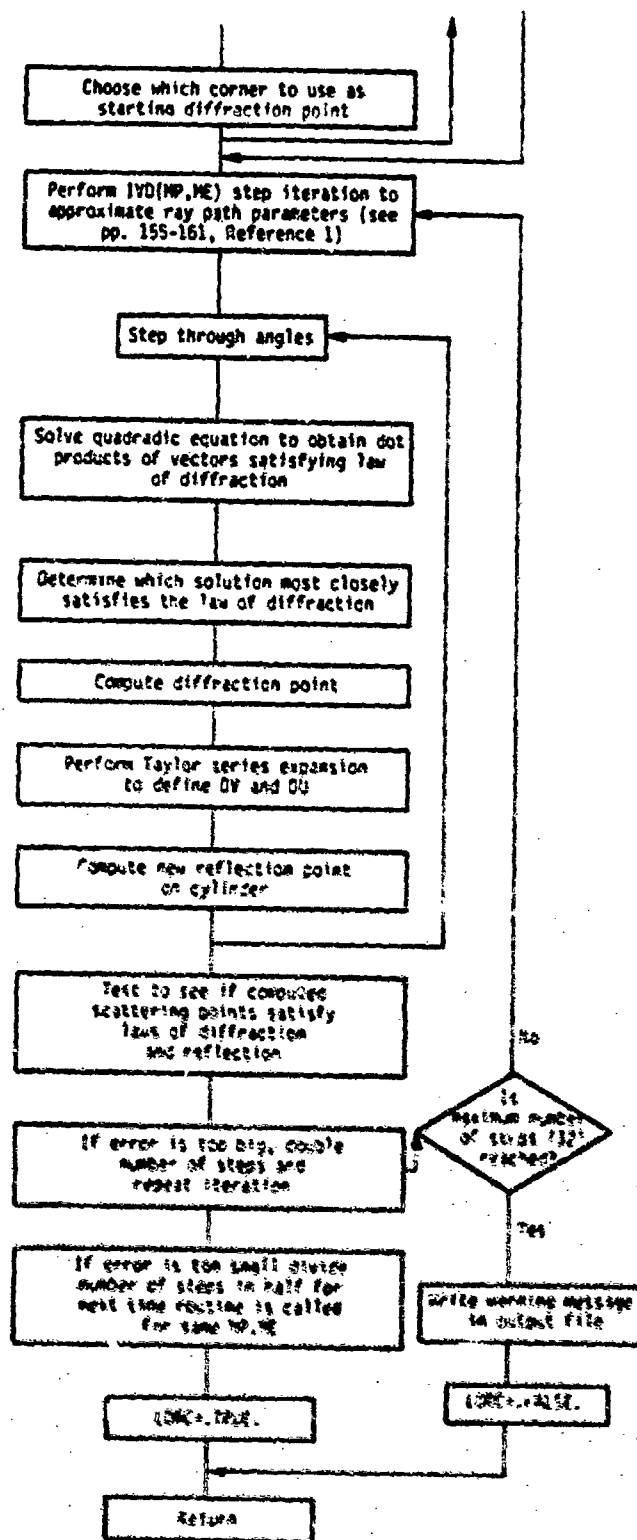
Figure 54--Ray diffracted by plate and then reflected by the cylinder.

METHOD

The diffraction point on a plate edge and the reflection point on an elliptic cylinder for a diffracted-reflected ray in a given observation direction are calculated via an iterative process. The equations are based on a first order Taylor series approximation to the equations governing the laws of reflection and diffraction. The details of the analysis are given on pages 155-161 of Reference 1. The iteration process follows the same basic scheme outlined in the write up for subroutine RFPTCL. The initial start up procedure for this subroutine is composed of defining a known reflection point which is taken to be on the rim of the finite cylinder closest to the plate edge under consideration and then determining the corresponding diffraction point on the plate edge. The details of this procedure are discussed on pages 161-163 of Reference 1.

FLOW DIAGRAM





SYMBOL DICTIONARY

CSCE	DOT PRODUCT OF RAY FROM CORNER OF EDGE ME TO SOURCE AND EDGE UNIT VECTOR
DPSM	PHI ANGLE INCREMENT SIZE
UR	REFLECTED RAY PROPAGATION DIRECTION
UMP	X,Y COMPONENTS OF PHI POLARIZATION UNIT VECTOR FOR FIELD REFLECTED FROM CYLINDER IN RCS
UMT	X,Y,Z COMPONENTS OF THETA POLARIZATION UNIT VECTOR FOR FIELD REFLECTED FROM CYLINDER
DTSM	THETA ANGLE INCREMENT SIZE
DU	CHANGE IN UR FOR ONE ITERATION USING TAYLOR SERIES EXPANSION
DV	CHANGE IN VR FOR ONE ITERATION USING TAYLOR SERIES EXPANSION
ENC	ERROR DETECTION VARIABLE
F1	EQUATION GOVERNING THE LAW OF REFLECTION
FP	PARTIAL DERIVATIVE OF F1 WITH RESPECT TO PHI
F _T	PARTIAL DERIVATIVE OF F1 WITH RESPECT TO THETA
FU	PARTIAL DERIVATIVE OF F1 WITH RESPECT TO UR
FV	PARTIAL DERIVATIVE OF F1 WITH RESPECT TO VR
G1	EQUATION GOVERNING THE LAW OF REFLECTION
GP	PARTIAL DERIVATIVE OF G1 WITH RESPECT TO PHI
GT	PARTIAL DERIVATIVE OF G1 WITH RESPECT TO THETA
GU	PARTIAL DERIVATIVE OF G1 WITH RESPECT TO UR
GV	PARTIAL DERIVATIVE OF G1 WITH RESPECT TO VR
IVD	NUMBER OF STEPS USED IN ITERATION
LDNC	SET TRUE IF STARTING POINT DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE
PHCR	PHI COMPONENT OF REFLECTED RAY DIRECTION
PHCK	PHI COMPONENT OF REFLECTED RAY DIRECTION FROM PREVIOUS TIME REFPT WAS CALLED (OR PRESENT VALUE FOR NEXT TIME ROUTINE IS CALLED)
PHCRP	PHI ANGLE OF REFLECTED RAY DIRECTION IN ROTATED RCS SYSTEM (BRANCH CUT PLACED BEHIND CYL)
PHSPH	PHI ANGLE OF REFLECTED RAY DIRECTION IN ROTATED RCS SYSTEM (BRANCH CUT PLACED BEHIND CYLINDER)
SNPX	PARTIAL DERIVATIVE OF SNX WITH RESPECT TO VR
SNPY	PARTIAL DERIVATIVE OF SNY WITH RESPECT TO VR
SNX	X AND Y COMPONENTS OF NORMAL TO CYLINDER IN RCS COMPONENTS
SNY	
SPD	NUMBER OF STEPS USED IN ITERATION
THCR	THETA COMPONENT OF REFLECTED RAY DIRECTION
THCK	THETA COMPONENT OF REFLECTED RAY DIRECTION FROM PREVIOUS TIME REFPT WAS CALLED (OR FOR NEXT TIME ROUTINE IS CALLED)
UNC	Z COMPONENT OF STARTING REFLECTION POINT LOCATION ON CYLINDER
VI	UNIT VECTOR OF INCIDENT RAY ON CYLINDER
VIG	PARTIAL DERIVATIVE OF VI WITH RESPECT TO UR
VIV	PARTIAL DERIVATIVE OF VI WITH RESPECT TO VR
VNC	ELL ANGLE DEFINING STARTING REFLECTION POINT ON CYLINDER
VSD	X,Y,Z COMPONENTS OF PROPAGATION VECTOR OF RAY FROM SOURCE TO DIFFRACTION POINT
XD	X,Y,Z COMPONENTS OF DIFFRACTION POINT LOCATION
XP	POINT ALONG LINE DRAWN THROUGH EDGE ME CLOSEST TO SOURCE
XN	X,Y,Z COMPONENTS OF REFLECTION POINT LOCATION ON CYLINDER
XND	PARTIAL DERIVATIVE OF XN WITH RESPECT TO UR
XNV	PARTIAL DERIVATIVE OF XN WITH RESPECT TO VR

CODE LISTING

```

1 C-----
2 SUBROUTINE DIFFPT(VR,XR,DOTP,SNZ,VIN,VI,XD,VSDP,VSD
3 2,DE,ME,MP,LDNC)
4 C!!!
5 C!!! DETERMINES THE RAY PATH FOR A DIFFRACTION FROM A PLATE THEN
6 C!!! A REFLECTION FROM AN ELLIPTIC CYLINDER
7 C!!!
8 DIMENSION DR(3),DMP(2),DHT(3),VI(3),VI/(3),VUC(3),VSD(3)
9 DIMENSION XP(3),XR(3),XHP(3),XHV(3),XHU(3),XD(3)
10 DIMENSION IVD(14,6),PHOR(14,6),THOR(14,6),VRO(14,6),URO(14,6)
11 DIMENSION PHCRP(14,6)
12 LOGICAL LDNC
13 COMMON/LEOPL//X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
14 2,MEP(14),MPA
15 COMMON/SORINF/XS(3),VXS(3,3)
16 COMMON/IRND(3),THSH,PHSH,SPHS,CPHS,STHS,CTHS
17 COMMON/LEONEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
18 COMMON/LEODCL/VDC(14,6),UDC(2),PHCR(14,6,2),TOCR(14,6,2)
19 2,DTDC(14,6),PTDC(14,6,4),DDC(14,6,2)
20 COMMON/LRNPRI/PHRI(14,6)
21 COMMON/PHIS/PI,TPI,DPR,RPD
22 C!!! PLACE BLANCH OUT BEHIND CYLINDER
23 PHSPR=PHSH-PI*(MP,1E)
24 IF (PHSPR.GT.PI) PHSPR=PHSPR-PI
25 IF (PHSPR.LT.-PI) PHSPR=PHSPR+PI
26 CSCE=.
27 DO 2 N=1,3
28 CSCE=CSCE+(XS(N)-X(MP,ME,N))*V(MP,ME,N)
29 SSN=.
30 DO 3 N=1,3
31 XP(N)=CSCE+V(MP,ME,N)*X(MP,ME,N)
32 SSN=SSN+(XS(N)-XP(N))*(XS(N)-XP(N))
33 SN=SQRT(SSN)
34 C!!! IS STARTING POINT DATA AVAILABLE FROM PREVIOUS
35 C!!! PATTERN ANGLE
36 IF (LDNC) GO TO 40
37 C!!! COMPUTE STARTING POINT
38 C!!! STEP THRU CORNERS ON EDGE ME AND CHOOSE
39 C!!! WHICH CORNER TO USE AS STARTING REF. POINT
40 CPDC=COS(PHCR(MP,ME,1))
41 SPDC=SIN(PHCR(MP,ME,1))
42 STDC=SIN(THCR(MP,ME,1))
43 DDC1=D(1)+CPDC*STDC*D(2)+SPDC*STDC*D(3)+DDC(MP,ME,1)
44 CPDC=COS(PHCR(MP,ME,2))
45 SPDC=SIN(PHCR(MP,ME,2))
46 STDC=SIN(THCR(MP,ME,2))
47 DDC2=D(1)+CPDC*STDC*D(2)+SPDC*STDC*D(3)+DDC(MP,ME,2)
48 IF (DDC2.GT.DDC1) JJ=2
49 PHCR(MP,ME)=PHCR(MP,ME,JJ)
50 THCR(MP,ME)=THCR(MP,ME,JJ)
51 PHOR(MP,ME)=PHOR(MP,ME,JJ)
52 IF (PHOR(MP,ME).GT.PI) PHOR(MP,ME)=PHOR(MP,ME)-PI
53 IF (PHOR(MP,ME).LT.-PI) PHOR(MP,ME)=PHOR(MP,ME)+PI
54 THOR(MP,ME)=THOR(MP,ME,JJ)
55 VDC(MP,ME)=VDC(MP,ME,JJ)
56 UDC(MP,ME)=UDC(MP,ME,JJ)
57 IVD(MP,ME)=1
58 C!!! SELECT NUMBER OF STEPS IN ITERATION
59 N=IVN(MP,ME)
60 IVM=1
61 PHOR(MP,ME)=PHOR(MP,ME)/STP
62 THOR(MP,ME)=THOR(MP,ME)/STP
63 C!!! COMPUTE STARTING POINT
64 X=XP(MP,ME)
65 Y=Y(MP,ME)
66 C!!! THROUGH IVN STEP ITERATIONS TO NUMERICALLY

```

```

07 CIII COMPUTE THE DIFFRACTION AND REFLECTION POINTS.
08 CIII STEP THRU ANGLES
09 DO 50 IV=1,IVDP
70 PHCH=PHCH(MP,ME)+(IV-1)*NPSH
71 THCH=THCH(MP,ME)+(IV-1)*DTSR
72 CPCS=COS(PHCH)
73 SPCS=SIN(PHCH)
74 CTCS=COS(THCH)
75 STCS=SIN(THCH)
76 DN(1)=CPCS*STCS
77 DN(2)=SPCS*STCS
78 DN(3)=CTCS
79 DNP(1)=-SPCS*STCS
80 DNP(2)=CPCS*STCS
81 DNT(1)=CPCS*CTCS
82 DNT(2)=SPCS*CTCS
83 DNT(3)=-STCS
84 CSV=COS(VR)
85 SNV=SIN(VR)
86 SHX=B*CSV
87 SHY=A*SNV
88 SHPX=-B*SNV
89 SHPY=A*CSV
90 XH(1)=A*CSV
91 XH(2)=B*SNV
92 XH(3)=UH
93 XHV(1)=-A*SNV
94 XHV(2)=E*CSV
95 XHV(3)=0.
96 XRU(1)=0.
97 XRU(2)=0.
98 XRU(3)=1.
99 CIII SOLVE QUADRATIC EQUATION TO OBTAIN DOT PRODUCT
100 CIII OF VECTORS, SATISFYING LAW OF DIFFRACTION
101 SSNP=0.
102 DO 10 N=1,3
103 XEP(N)=XRP(N)-XPN(N)
104 SSNP=SSNP+XEP(N)*XRP(N)
105 CRPV=XRP(1)*V(MP,ME,1)+XRP(2)*V(MP,ME,2)+XRP(3)*V(MP,ME,3)
106 AA=(SSNP-SSN)*(SSNP-SSN)+4.*SSN*CRPV*CRPV
107 BB=-2.*(SSN*SSNP)+CRPV*CRPV
108 CC=CRPV*CRPV*CRPV*CRPV
109 SQBAC=SQRT((BB*BB-4.*AA*CC))
110 CIII DETERMINE WHICH SOLUTION MOST CLOSELY SATISFIES
111 CIII THE LAW OF DIFFRACTION
112 ANSA=(-BB-SQBAC)/2./AA
113 ANSB=(-BB+SQBAC)/2./AA
114 CCIV=ANSA
115 IF (ABS(ANSA-1.0).GT.(ABS(ANSA-1.0))) CCIV=ANSB
116 CIII CIV=SQRT(CCIV)
117 JCIV=0
118 14 JCIV=JCIV+1
119 VSDM=0.
120 VSDM=0.
121 DO 11 N=1,3
122 CIII COMPUTE DIFFRACTION POINT
123 XDP(N)=XPN(N)+SPCIV*V(MP,ME,N)/SQRT(1.-CCIV)
124 VSDN(XDP(1)-XPN(1))
125 VSDN(XDP(2)-XPN(2))
126 VSDN(XDP(3)-XPN(3))
127 11 VDP(V)=VDP(V)+VSDN(V)
128 VDP(V)=SQRT(VDP(V))
129 VSDN=VSDN/VDP(V)
130 VC=VDP(1)/V(MP,ME,1)+VDP(2)/V(MP,ME,2)+VDP(3)/V(MP,ME,3)
131 FSDN=VSDN(1)*V(MP,ME,1)+VSDN(2)*V(MP,ME,2)+VSDN(3)*V(MP,ME,3)
132 21/VSDN.

```

```

132 ERCE=1E-150
133 ERCE=ABS(ERCE)
134 IF(ERCE.LT.0.01) GO TO 15
135 CIV=-CIV
136 IF(JCIV.LT.2) GO TO 14
137 15 CONTINUE
138 IF(IV.EQ.IVDP) GO TO 66
139 C!!! PERFORM TAYLOR SERIES EXPANSION TO DEFINE DV AND DU
140 CXRV=XRV(1)*V(MP,ME,1)+XRV(2)*V(MP,ME,2)+XRV(3)*V(MP,ME,3)
141 CXRV=(CXRV(1)*VI(1)+XRV(2)*VI(2)+XRV(3)*VI(3))/VIM
142 CIVE=(CXRV-CXRV*CIV)/(VIM+SM/SCRT(1.-CCIV))
143 CXRU=XRU(1)*V(MP,ME,1)+XRU(2)*V(MP,ME,2)+XRU(3)*V(MP,ME,3)
144 CXRU=(CXRU(1)*VI(1)+XRU(2)*VI(2)+XRU(3)*VI(3))/VIM
145 CUIE=(CXRU-CXRU*CIV)/(VIM+SM/SCRT(1.-CCIV))
146 DO 12 N=1,3
147 VIV(N)=SM*CIVE*(1.+CCIV/(1.-CCIV))/SCRT(1.-CCIV)
148 VIU(N)=XRV(N)-VIV(N)*V(MP,ME,N)
149 VIU(N)=SM*CUIE*(1.+CCIV/(1.-CCIV))/SCRT(1.-CCIV)
150 12 VIU(N)=XRU(N)-VIU(N)*V(MP,ME,N)
151 FV=(SNX*VI(1)+SNX*VIV(1)+SNPY*VI(2)+SNPY*VIV(2))*
152 2(SNX*DR(2)-SNPY*DR(1))
153 FV=FV+(SNX*VI(1)+SNPY*VI(2))*(SNPX*DR(2)-SNPY*DR(1))
154 FV=FV+(SNPX*VI(2)+SNX*VIV(2)-SNPY*VI(1)-SNPY*VIV(1))*
155 2(SNX*DR(1)+SNPY*DR(2))
156 FV=FV+(SNX*VI(2)-SNPY*VI(1))*(SNPX*DR(1)+SNPY*DR(2))
157 FH=(SNX*DR(2)-SNPY*DR(1))*(SNX*VIU(1)+SNPY*VIU(2))+
158 2(SNX*DR(1)+SNPY*DR(2))*(SNX*VIU(2)-SNPY*VIU(1))
159 GV=DR(3)*(SNX*VI(1)+SNX*VIV(1)+SNPY*VI(2)+SNPY*VIV(2))
160 GV=GV+VI(3)*(SNPX*DR(1)+SNPY*DR(2))
161 GV=GV+VIV(3)*(SNX*DR(1)+SNPY*DR(2))
162 GU=DR(3)*(SNX*VIU(1)+SNPY*VIU(2))+VIU(3)*(SNX*DR(1)+SNPY*DR(2))
163 FP=(SNX*VI(1)+SNPY*VI(2))*(SNX*DRP(2)-SNPY*DRP(1))+
164 2(SNX*VI(2)-SNPY*VI(1))*(SNX*DRP(1)+SNPY*DRP(2))
165 GP=VI(3)*(SNX*DRP(1)+SNPY*DRP(2))
166 GT=DRT(3)*(SNX*VI(1)+SNPY*VI(2))+VI(3)*(SNX*DRT(1)+SNPY*DRT(2))
167 FI=(SNX*VI(1)+SNPY*VI(2))*(SNX*DR(2)-SNPY*DR(1))+
168 2(SNX*VI(2)+SNPY*DR(2))*(SNX*VI(2)-SNPY*VI(1))
169 GI=DR(3)*(SNX*VI(1)+SNPY*VI(2))+VI(3)*(SNX*DR(1)+SNPY*DR(2))
170 DET=FU*GV-FV*GU
171 DV=((FI*GU-GI*FU)+(GU*FP-FU*GP)*DPSR-FU*GT*DTSR)/DET
172 DU=((GI*FV-FI*GV)+(FV*GP-GV*FP)*DPSR+FV*GT*DTSR)/DET
173 CCM COMPUTE NEW REFLECTION POINT ON CYLINDER
174 UR=UR+DU
175 VR=VR+DV
176 40 VV=VR+DV
177 10 CONTINUE
178 60 CONTINUE
179 C!!! TEST TO SEE IF COMPUTED SCATTER POINTS SATISFY
180 C!!! LAWS OF DIFFRACTION AND REFLECTION
181 SNM=SQRT(SNX*SNX+SNY*SNY)
182 SNX=SNX/SNM
183 SNY=SNY/SNM
184 DO 20 N=1,3
185 VSD(N)=VSD(N)/VSDM
186 20 VI(N)=VI(N)/VIM
187 SHAD=SNX*DI(1)+SNY*DI(2)
188 SHAD=SHX*VI(1)+SNY*VI(2)
189 ENC=SHAD+SHADC
190 DCTP=.5*(SHAD-SHADC)
191 ERCA=ABS(ERCA)
192 ERCE=ERCE
193 IF(ERCE.GT.ERCE)ERCE=ERCE
194 C!!! IF ERROR IS VERY SMALL, CUT NUMBER OF ITERATIONS
195 C!!! IN HALF FOR NEXT TIME ROUTINE IS CALLED
196 IF(ERCE.LT.0.01) GO TO 80
197 C!!! IF ERROR IS TOO BIG, DOUBLE NUMBER OF INCREMENTS
198 C!!! (UP TO 32) AND REPEAT ITERATION

```

```

199      IF(IVD(MP,ME).GE.32) GO TO 70
200      IVD(MP,ME)=2*IVD(MP,ME)
201      GO TO 40
202 70    CONTINUE
203      WRITE(6,1) PHSR,THSR,MP,ME,VR,UR,ERCA,ERCR
204 1     FORMAT(' ERROR IN DFRFPT= ',2F12.6,2I5,4F12.6)
205      LDRC=.FALSE.
206      RETURN
207 40    CONTINUE
208      IF(ERC.GE.0.001) GO TO 90
209      IF(IVD(MP,ME).EQ.1) GO TO 90
210      IVD(MP,ME)=IVD(MP,ME)/2
211 90    CONTINUE
212 C!!!  STOKE PARAMETERS FOR NEXT TIME DFRFPT IS CALLED
213      VRO(MP,ME)=VR
214      URO(MP,ME)=UR
215      PHOR(MP,ME)=PHSR
216      PHORP(MP,ME)=PHSPR
217      THOR(MP,ME)=THSR
218      IF(.NOT.LDRC) IVD(MP,ME)=1
219      LDRC=.TRUE.
220      RETURN
221      END

```

DI

PURPOSE

To calculate the incident part or the reflection part of the wedge diffraction coefficient or the corner diffraction coefficient.

METHOD

This subroutine computes either the incident part or the reflection part of the wedge or corner diffraction coefficient. The uniform Geometrical Theory of Diffraction [4] has been used to derive these terms. For wedge diffraction the coefficient is given as

$$DI(R, \beta, \sin \beta_0, n) = \frac{-e^{-j\pi/4}}{2n\sqrt{2\pi k} \sin \beta_0} \left\{ \cot\left(\frac{\pi+\beta}{2n}\right) F[kRa^+(\beta)] + \cot\left(\frac{\pi-\beta}{2n}\right) F[kRa^-(\beta)] \right\},$$

where $\beta = \begin{cases} \phi - \phi', & \text{for the incident case} \\ \phi + \phi', & \text{for the reflection case,} \end{cases}$

$$a_{\pm}^{\pm}(\beta) = 2 \cos^2 \left(\frac{2\pi n N_{\pm}^{\pm} - \beta}{2} \right),$$

in which $N_{\pm}^{\pm}$ are the integers which most nearly satisfy the equations

$$2\pi n N^+(\beta) = \pi$$

$$2\pi n N^-(\beta) = -\pi,$$

$F(x)$ is the transition function,

and

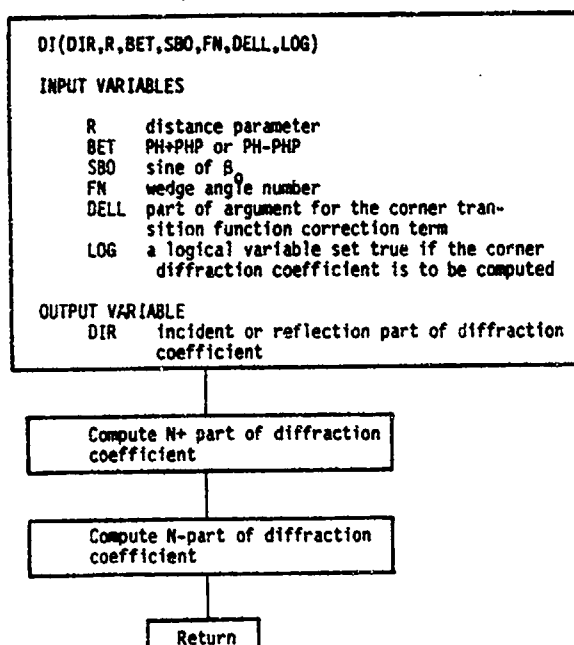
n is the wedge number (FN).

For the corner diffracted term (LOG=.TRUE.), the coefficient is given as [9]:

$$DI(R, \beta, \sin \beta_0, n, R_c) = \frac{-e^{-j\pi/4}}{2n\sqrt{2\pi k} \sin \beta_0} \left\{ \cot\left(\frac{\pi+\beta}{2n}\right) F[kRa^+(\beta)] \times \left| F\left[\frac{Ra^+(\beta)/\lambda}{kR_c a(\pi+\beta_0-\beta_c)}\right] \right| + \cot\left(\frac{\pi-\beta}{2n}\right) F[kRa^-(\beta)] \left| F\left[\frac{Ra^-(\beta)/\lambda}{kR_c a(\pi+\beta_0-\beta_c)}\right] \right| \right\},$$

where R_c is the corner distance parameter and β_c is the theta type angle measured from the corner. An illustration of the geometry is given in Figure 55.

FLOW DIAGRAM



SYMBOL DICTIONARY

A	ANGULAR FUNCTION FOR TRANSITION FUNCTION
ANG	BET IN RADIANS
BOTL	ARGUMENT OF TRANSITION FUNCTION
C	REAL PART OF FRESNEL INTEGRAL
COM	CONSTANT FOR DIFFRACTION COEFFICIENT
COTA	COTANGENT TIMES THE SQUARE ROOT OF THE A FUNCTION
DEL	CORNER PART OF ARGUMENT FOR THE CORNER TRANSITION FUNCTION CORRECTION TERM
DELU	INVERSE OF DEL
DEK	$4\pi \cdot FN \cdot \sin(\theta_0)$
DN	INTEGER WHICH MOST NEARLY SATISFIES THE EQUATION, $2\pi \cdot FN \cdot DN - BET = \pi$ OR $-\pi$
DNS	COMPUTATIONAL VARIABLE
EX	$\exp(j \cdot k \cdot R \cdot A)$
FA	TRANSITION FUNCTION WITHOUT SORT(A)
N	COMPUTATIONAL VARIABLE
HAG	ARGUMENT OF COTANGENT TERM
S	IMAGINARY PART OF FRESNEL INTEGRAL
SGN	SIGN OF DNS
SQR	$\sqrt{2\pi \cdot R}$
TS	ABSOLUTE VALUE OF TSIN
TSIN	SINE OF ARGUMENT OF COTANGENT TERM
UNPI	N- COMPONENT OF DI
UPPI	N+ COMPONENT OF DI

CODE LISTING

```

1 C-----
2 SUBROUTINE DI(DIR,R,BET,SBO,FN,DELL,LOG)
3 C!!!
4 C!!! INCIDENT (BET=PH-PHP) OR REFLECTED (BET=PH+PHP)
5 C!!! PART OF WEDGE DIFFRACTION COEFFICIENT
6 C!!!
7 LOGICAL LOG,LDEBUG,LTEST
8 COMMON/TEST/LDEBUG,LTEST
9 COMPLEX FFCT, TOP, COM, EX, UPPI, UNPI, FA, DIR
10 COMMON/TOPD/TOP
11 COMMON/PIS/PI, TPI, DPR, RPD
12 IF (LDEBUG) WRITE (6,11)
13 11 FORMAT (/, ' DEBUGGING DI SUBROUTINE')
14 DEL=DELL
15 IF (ABS(DELL).LT.1.E-10) DEL=SIGN(1.E-10,DEL)
16 IF (LOG) DELU=1./DEL
17 ANG=BET*RPD
18 DEM=2.*TPI*FN*SBO
19 COM=TOP/DEM
20 SQR=SQRT(TPI*R)
21 C!!! N+ PART OF DIFFRACTION COEFFICIENT
22 DNS=(PI+ANG)/(2.*FN*PI)
23 SGN=SIGN(1.,DNS)
24 N=IFIX(ABS(DNS)+0.5)
25 DN=SGN*FLOAT(N)
26 A=ABS(1.0+COS(ANG-2.*FN*PI*DN))
27 BOTL = 2.*SQRT(ABS(R*A))
28 EX=CEXP(CMPLX(0.0,TPI*R*A))
29 CALL FRNELS (C,S,BOTL)
30 C=SQRT(PI/2.0)*(0.5-C)
31 S= SQRT(PI/2.0)*(S-0.5)
32 FA=CMPLX(0.,2.)*SQR*EX*CMPLX(C,S)
33 RAG=(PI+ANG)/(2.*FN)
34 TSIN=SIN(RAG)
35 TS=ABS(TSIN)
36 IF (TS.GT.1.E-5) GO TO 442
37 COTA=-SQRT(2.0)*FN*SIN(ANG/2.0-FN*PI*DN)
38 IF (COS(ANG/2.0-FN*PI*DN).LT.0.0) COTA=-COTA
39 GO TO 443
40 442 COTA=SQRT(A)*COS(RAG)/TSIN
41 443 UPPI=COM*COTA*FA
42 IF (LOG) UPPI=UPPI*BABS(FFCT(R*A*DELU))
43 IF (LDEBUG) WRITE (6,*) DN,A,FA,UPPI
44 C!!! N- PART OF DIFFRACTION COEFFICIENT
45 DNS=(-PI+ANG)/(2.*FN*PI)
46 SGN=SIGN(1.,DNS)
47 N=IFIX(ABS(DNS)+0.5)
48 DN=SGN*FLOAT(N)
49 A=ABS(1.0+COS(ANG-2.*FN*PI*DN))
50 BOTL = 2.*SQRT(ABS(R*A))
51 EX=CEXP(CMPLX(0.0,TPI*R*A))
52 CALL FRNELS (C,S,BOTL)
53 C=SQRT(PI/2.0)*(0.5-C)
54 S= SQRT(PI/2.0)*(S-0.5)
55 FA=CMPLX(0.,2.)*SQR*EX*CMPLX(C,S)
56 RAG=(PI-ANG)/(2.*FN)
57 TSIN=SIN(RAG)
58 TS=ABS(TSIN)
59 IF (TS.GT.1.E-5) GO TO 542
60 COTA= SQRT(2.0)*FN*SIN(ANG/2.0-FN*PI*DN)
61 IF (COS(ANG/2.0-FN*PI*DN).LT.0.0) COTA=-COTA
62 GO TO 123
63 542 COTA=SQRT(A)*COS(RAG)/TSIN
64 123 UNPI=COM*COTA*FA
65 IF (LOG) UNPI=UNPI*BABS(FFCT(R*A*DELU))

```

```

66      IF (LDEEUG) WRITE (6,*) DN,A,FA,UNPI
67      DIR=UPPI+UNPI
68      IF (.NOT.LTEST) GO TO 2
69      WRITE (6,1)
70 1     FORMAT (/,' TESTING DI SUBROUTINE')
71      WRITE (6,*) DIR,R,BET
72      WRITE (6,*) SBO,FN
73 2     RETURN
74      END

```

DIFPLT

PURPOSE

To calculate the far zone electric field for a source ray which is diffracted off of a given edge on a given plate.

PERTINENT GEOMETRY

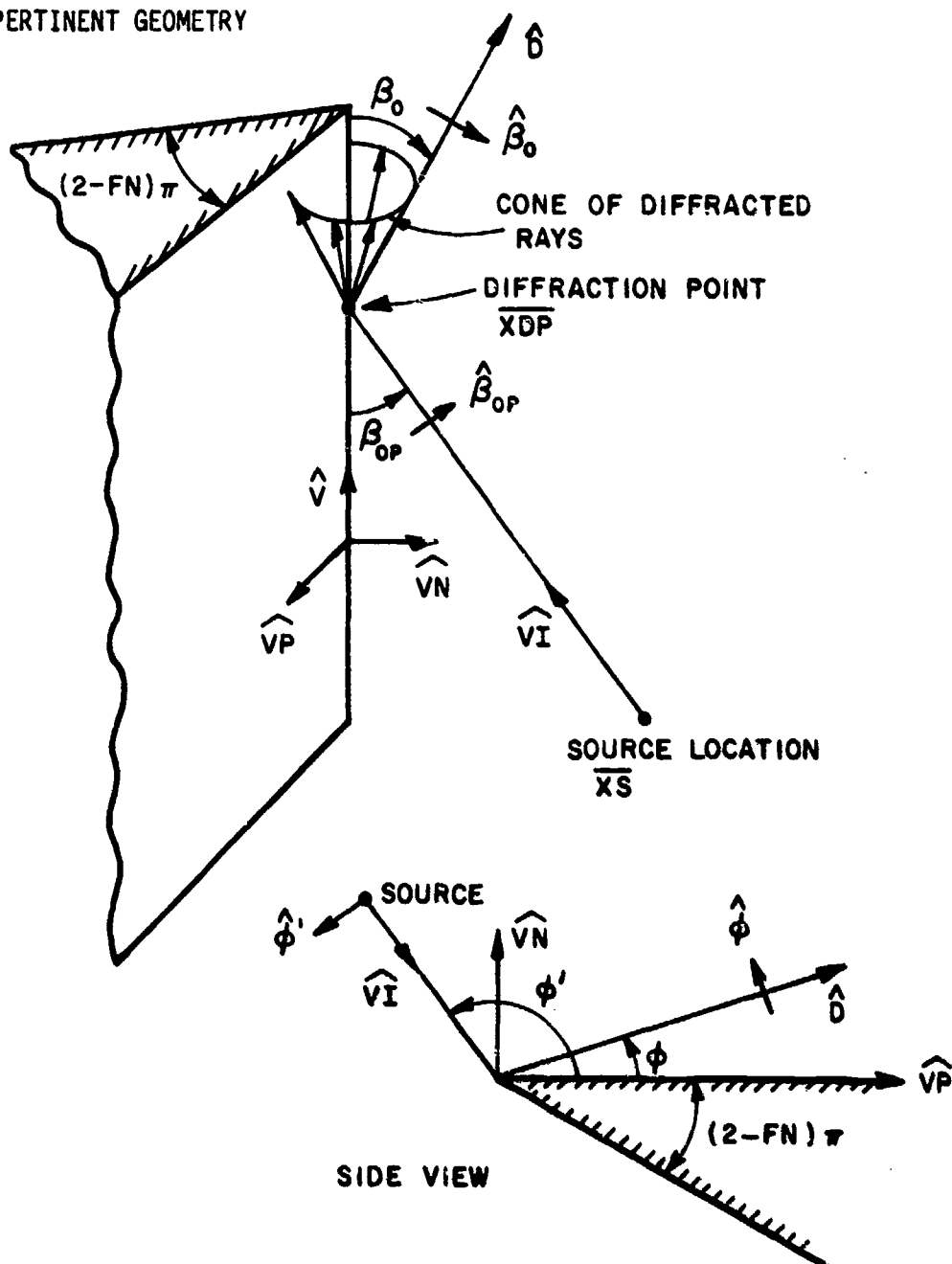


Figure 55--Edge diffraction geometry.

$$\hat{\beta}_0 = \hat{x} B0(1) + \hat{y} B0(2) + \hat{z} B0(3)$$

$$\hat{\beta}_{op} = \hat{x} BOP(1) + \hat{y} BOP(2) + \hat{z} BOP(3)$$

$$\hat{\phi} = \hat{x} PH(1) + \hat{y} PH(2) + \hat{z} PH(3)$$

$$\hat{\phi}' = \hat{x} PHO(1) + \hat{y} PHO(2) + \hat{z} PHO(3)$$

$$\phi' = \text{PSOR}$$

$$\phi = \text{PSR}$$

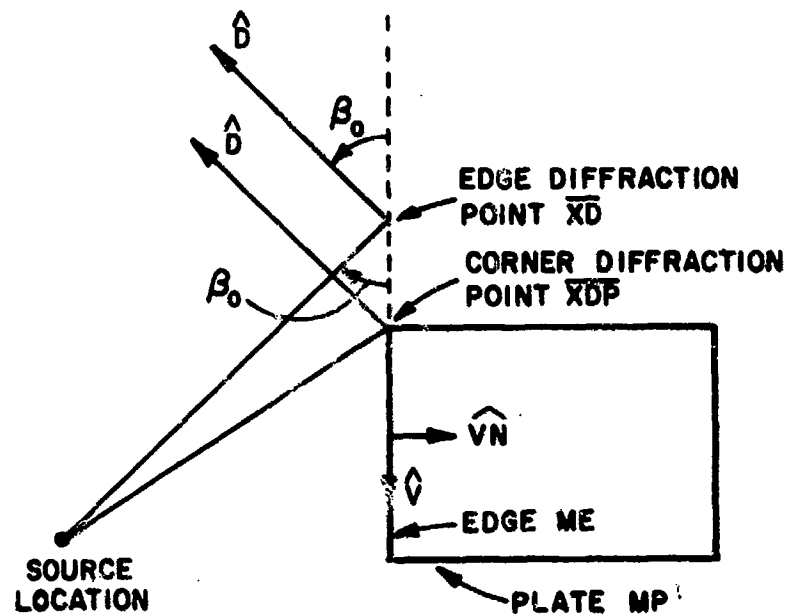


Figure 56--Corner diffraction geometry.

METHOD

The diffracted fields from the edges of the plates are calculated by using the Geometrical Theory of Diffraction [4]. The diffracted field in the far zone has the form [4]

$$\vec{E}^d = \vec{E}^i(Q_E) \cdot \vec{D}_E(s', \phi, \phi', \beta_0, R) \sqrt{s'} \frac{e^{-jks}}{s},$$

where Q_E is the diffraction point. The incident field can be written in the form

$$\vec{E}^i(Q_E) = [EIPR \hat{\phi}' + EIPL \hat{\beta}_{op}] \frac{e^{-jks'}}{s'}.$$

The diffraction coefficient can be written as:

$$\bar{D}_E(s', \phi, \phi', \beta_0, FN) = -DS \beta_{op} \beta_0 - DH \hat{\phi}' \hat{\phi}.$$

The slope diffracted field in the far zone has the form[10]

$$E^{s.d.} = \frac{1}{jk \sin \beta_0} \frac{\partial \bar{E}^i(Q_E)}{\partial n} \cdot \frac{\partial \bar{D}_E}{\partial \phi'} \sqrt{s'} \frac{e^{-jks}}{s}$$

where $\frac{\partial \bar{E}^i}{\partial n} = \frac{1}{s' \sin \beta_0} \frac{\partial \bar{E}^i}{\partial \phi'}$. The incident slope field can be written

in the form $\frac{\partial \bar{E}^i}{\partial n} = [EIPRP \hat{\phi}' + EIPLP \beta_{op}] \frac{e^{-jks'}}{s'^2}$ where EIPRP and EIPLP

are computed in subroutine SOURCP. The corner and slope corner diffracted fields have similar form[9] and are included if the logical variables LSLOPE and LCORNR are set true. The edge and slope fields are combined and the phase is referred to the reference coordinate system origin by the factor $e^{jkD \cdot \hat{XDP}}$. The form of the field is therefore given by

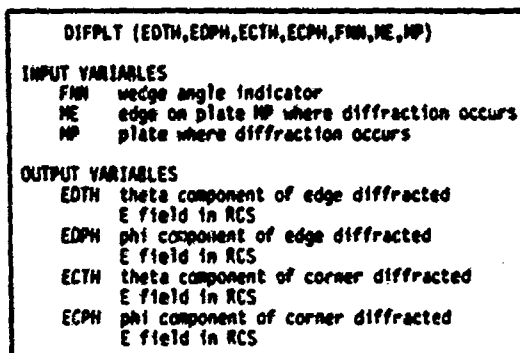
$$E^d = W_m (EDTH\hat{\theta} + EDPH\hat{\phi}) \frac{e^{-jkR}}{R}.$$

Similarly the corner and slope corner diffracted field is given by

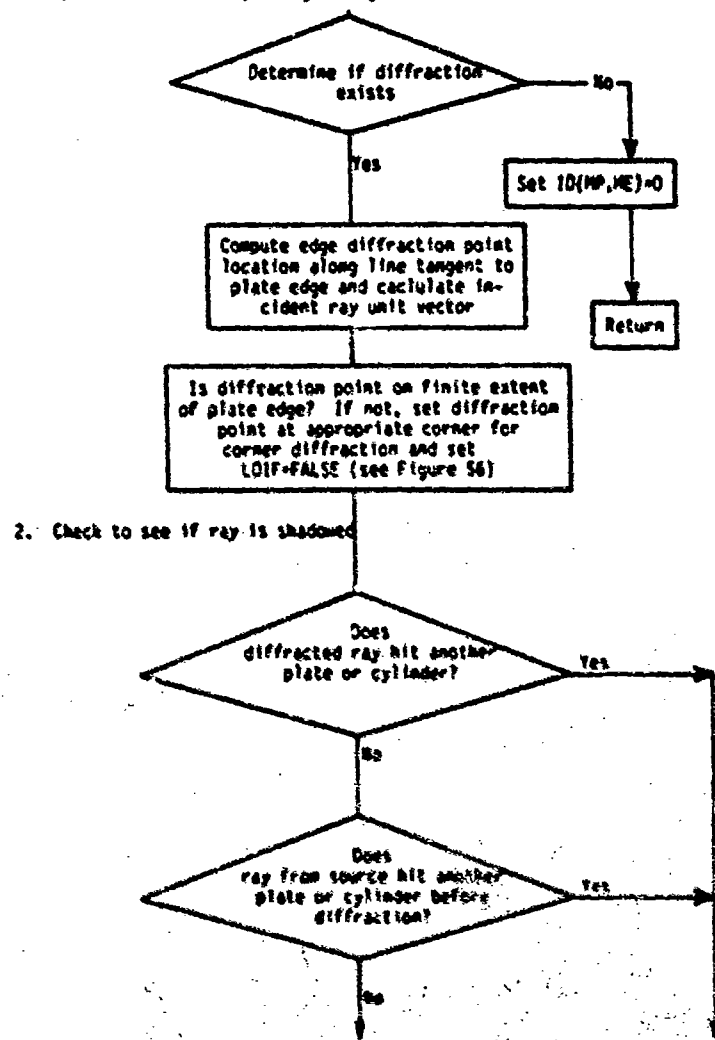
$$E^c = W_m (ECTH\hat{\theta} + ECPH\hat{\phi}) \frac{e^{-jkR}}{R},$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM



1. Perform diffraction point geometry calculations



3. Calculate diffraction angles and related geometry.

Calculate PSD and PS, the
incident and diffracted
phi angles

Check to see if diffractions
just appeared. If so, set
flag in IDG(IANG) for use
in double diffraction.
Set ID(MP,ME)=-1

Compute diffraction polarization
unit vectors (PH,PHD,SD,SDP)

Calculate $SDG = \sin(\alpha_0)$

4. Calculate edge diffracted fields

Compute source field pattern factor

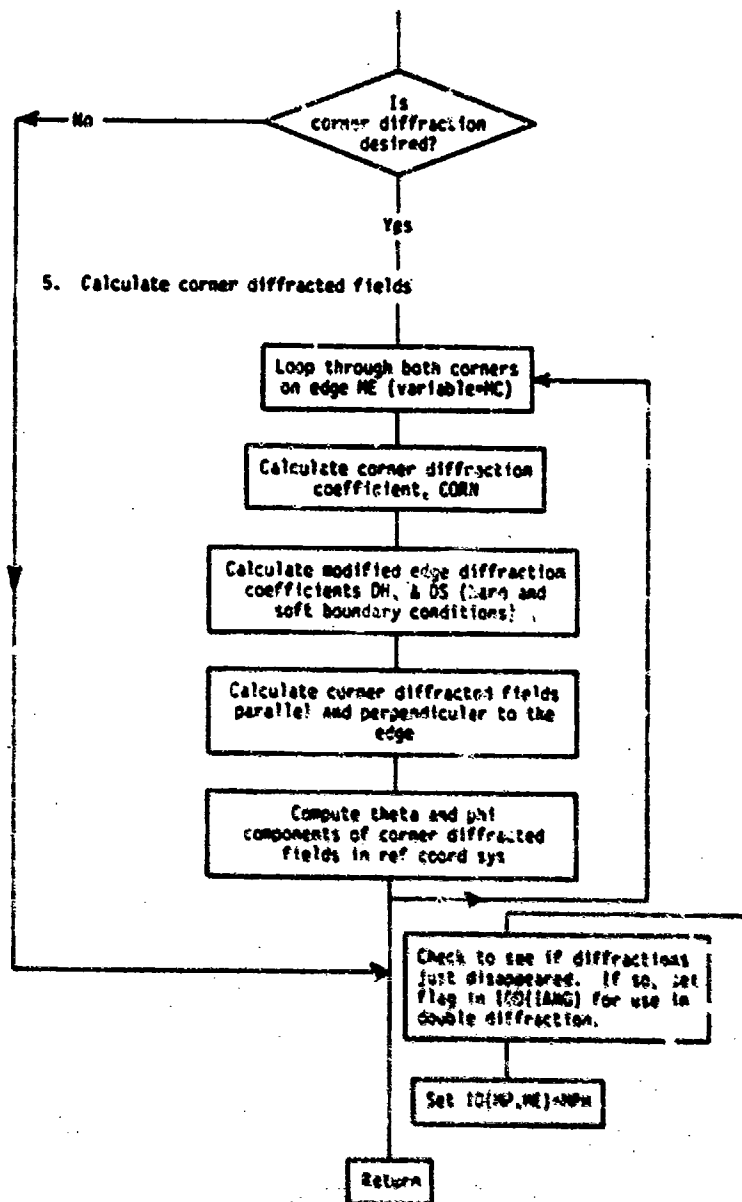
If slope diffraction is desired,
compute incident slope field
pattern factor

Calculate phase term
(refer phase to origin)

Compute edge diffraction coefficients

Compute perpendicular and
parallel components of
diffracted field (SDPR,SDPL)

If edge diffraction exists, (SDPR≠0)
compute theta and phi components of
the diffracted field in RCS



SYMBOL DICTIONARY

ADN	DOT PRODUCT OF VECTOR FROM PLATE MP TO THE SOURCE AND THE PLATE UNIT NORMAL
AFN	WEDGE ANGLE NUMBER
SUCL	VARIABLE USED TO EXPAND DIFFRACTION ANGLE RANGE IF CORNER DIFFRACTION IS USED
BDHI	UPPER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA
BDLCN	LOWER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA
SETN	DIFFERENCE IN DIFFRACTED AND INCIDENT PHI ANGLES
BETP	SUM OF DIFFRACTED AND INCIDENT PHI ANGLES
BO	DIFFRACTED FIELD BETA POLARIZATION UNIT VECTOR (IN EDGE FIXED COORD SYS) IN RCS COMPONENTS
BOP	INCIDENT FIELD BETA POLARIZATION UNIT VECTOR (IN EDGE FIXED COORD SYS) IN RCS COMPONENTS
CNP	COSINE OF HALF WEDGE ANGLE
CONN	CORNER DIFFRACTION COEFFICIENT
CPH	COSINE OF PSN
CPHC	COSINE OF PSCH
CTH	COSINE OF THN
CTHP	COSINE OF THPH
DEL	PARAMETER USED IN TRANSITION FUNCTION
DH	DIFFRACTION COEF. FOR HARD BOUNDARY CONDITION
DHIT	DISTANCE FROM SOURCE TO NEAREST HIT (FROM SUBS. PLAIN OR CYLIND)
DPH	SLOPE DIFFRACTION COEFFICIENT FOR HARD BOUNDARY CONDITION
DPS	SLOPE DIFFRACTION COEFFICIENT FOR SOFT BOUNDARY CONDITION
DS	DIFFRACTION COEF. FOR SOFT BOUNDARY CONDITION
DV	DOT PRODUCT OF EDGE VECTOR AND PROPAGATION DIRECTION UNIT VECTOR, D WHICH IS THE COSINE OF BETA
ECBI	EDGE DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR INCIDENT DIFFRACTED FIELD MODIFIED FOR CORNER DIFFRACTION
ECBN	EDGE DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR REFLECTED DIFFRACTED FIELD MODIFIED FOR CORNER DIFFRACTION
ECPH	PHI COMPONENT OF CORNER DIFFRACTED E-FIELD
ECTH	THETA COMPONENT OF CORNER DIFFRACTED E-FIELD
EDPH	PHI COMPONENT OF EDGE DIFFRACTED E-FIELD
EDPL	COMPONENT OF DIFFRACTED FIELD PARALLEL TO THE EDGE
EDPN	COMPONENT OF DIFFRACTED FIELD PERPENDICULAR TO THE EDGE
EDTH	THETA COMPONENT OF EDGE DIFFRACTED E-FIELD
ET	THETA COMPONENT OF CORNER DIFFRACTED FIELD IN RCS
EO	PHI COMPONENT OF CORNER DIFFRACTED FIELD IN RCS
EIPL	COMPONENT OF INCIDENT FIELD PARALLEL TO THE EDGE
EIPLD	PATTERN FACTOR FOR COMPONENT OF SOURCE (INCIDENT) SLOPE FIELD PARALLEL TO THE EDGE
EIPN	COMPONENT OF INCIDENT FIELD PERPENDICULAR TO THE EDGE
EIPND	PATTERN FACTOR FOR COMPONENT OF SOURCE (INCIDENT) SLOPE FIELD PERPENDICULAR TO THE EDGE
EIX	SOURCE PATTERN FACTORS FOR X, Y, AND Z COMPONENTS OF INCIDENT E FIELD
EIY	
EIZ	
EXPH	COMPLEX PHASE TERM (REFER PHASE TO RCS. ORIGIN)
FA	WEDGE ANGLE NUMBER
FIN	WEDGE ANGLE INDICATOR
FND	ANGLE EXTERIOR TO WEDGE IN DEGREES
GAN	DOT PRODUCT OF THE PROPAGATION DIRECTION AND THE VECTOR FROM THE REF COORD SYS ORIGIN TO THE DIFFRACTION POINT
ID	ARRAY OF FLAGS INDICATING WHETHER OR NOT DIFFRACTION WAS PRESENT THE LAST TIME DIFFRA WAS CALLED FOR EDGE NE OF PLATE MP (ID=-1 INDICATES DIFFRACTION PRESENT)
IDU	DOUBLE DIFFRACTION SHADOW BOUNDARY IDENTIFICATION ARRAY
ISN	SIGN CHANGE VARIABLE
UNIT	SET TRUE IF RAY HITS A PLATE OR CYLINDER (FROM PLAIN OR CYLIND)
NC	CORNER AT END OF EDGE NE
NE	EDGE ON PLATE MP WHERE DIFFRACTION OCCURS
NP	PLATE FOR WHICH DIFFRACTION OCCURS
N	DO LOOP VARIABLE
PO	DOT PRODUCT OF EDGE BISECTOR AND DIFF RAY PROPAGATION DIR

PH DIFFRACTED FIELD PHI POLARIZATION UNIT VECTOR (IN EDGE
 FIXED COORD SYS) IN RCS COMPONENTS
 PHIR PHI COMPONENT OF INCIDENT RAY PROPAGATION DIR IN RCS
 PHO INCIDENT FIELD PHI POLARIZATION UNIT VECTOR (IN EDGE
 FIXED COORD SYS) IN RCS COMPONENTS
 PHSR PHI COMPONENT OF DIF RAY PROPAGATION DIRECTION IN RCS
 PP NEGATIVE DOT PRODUCT OF EDGE BINORMAL AND INCIDENT RAY
 PROPAGATION DIRECTION
 PS PSR*DPH
 PSD DIFFRACTED RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM
 PSO PSOR*DPH
 PSOD INCIDENT RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM
 PSOR PHI COMPONENT OF INCIDENT RAY DIRECTION IN EDGE
 FIXED COORDINATE SYSTEM
 PSR PHI COMPONENT OF DIFFRACTED RAY PROPAGATION DIRECTION
 IN EDGE-FIXED COORDINATE SYSTEM
 QD DOT PRODUCT OF PLATE NORMAL AND DIF RAY PROPAGATION DIR
 QI NEGATIVE OF DOT PRODUCT OF PLATE NORMAL AND INCIDENT RAY
 PROPAGATION DIRECTION
 RM MAGNITUDE OF VECTOR FROM CORNER MC TO SOURCE
 RX }
 RY } X, Y, AND Z COMPONENTS OF VECTOR FROM CORNER MC TO SOURCE
 RZ }
 SBO SINE OF BO, THE ANGLE THE DIFFRACTED RAY MAKES WITH
 THE EDGE UNIT VECTOR
 SNP SINE OF HALF WEDGE ANGLE
 SP DISTANCE FROM SOURCE TO DIFFRACTION POINT (FROM SUB. DFPTND)
 SPH SINE OF PSR
 SPHO SINE OF PSOR
 SPP DISTANCE FROM SOURCE TO DIFFRACTION POINT
 SIH SINE OF THH
 TERM COEFFICIENT OF CORNER DIFFRACTED FIELDS
 THIR THETA COMPONENT OF INCIDENT RAY DIRECTION IN REF COORD SYS
 THPH ANGLE DIFFRACTED RAY MAKES WITH EDGE
 THH ANGLE BETWEEN EDGE UNIT VECTOR AND RAY FROM SOURCE
 TO CORNER MC
 TPP DISTANCE PARAMETER USED IN CALCULATING DIFFRACTION COEFFICIENTS
 VECT VECTOR USED TO MOVE DIFFRACTION POINT OFF EDGE FOR
 SHADOWING TESTS
 VI UNIT VECTOR OF INCIDENT RAY PROPAGATION DIR (FROM SUB. DFPTND)
 VIP UNIT VECTOR FROM SOURCE TO DIFFRACTION POINT
 VMO DISTANCE ALONG THE EDGE FROM FIRST CORNER OF EDGE TO DIF
 POINT
 VXS 3X3 MATRIX DEFINING THE SOURCE COORDINATE SYSTEM AXES
 XD DIFFRACTION POINT (CALCULATED IN SUB. DFPTND)
 XDP DIFFRACTION POINT (USED FOR SHADOWING TESTS)
 XS SOURCE LOCATION IN REF COORD SYS
 ZP DOT PRODUCT OF DIFFRACTED RAY PROPAGATION DIRECTION
 UNIT VECTOR D AND VECTOR FROM DIF POINT TO CORNER MC

CODE LISTING

```

1 C-----
2 SUBROUTINE DIFPLT(EDTH,EDPH,ECTH,ECPH,FKN,VE,MP)
3 C!!!
4 C!!! DETERMINES THE DIFFRACTED FIELD FROM EDGE #ME ON PLATE #MP
5 C!!! WITH THE PHASE REFERRED TO ORIGIN.
6 C!!! SLOPE AND CORNER DIFF. IS OPTIONAL FROM INPUT DATA.
7 C!!!
8 COMPLEX EF,EG,EIPR,EIPL,EXPH,EDPR,EDOL,EDTH,EDPH
9 COMPLEX ECTH,ECPH,ECRI,ECSR,DS,DH,DPS,DPH
10 COMPLEX EIPHP,EIPLP,EIX,EIY,EIZ,CORN,PFCT
11 DIMENSION VI(3),XD(3),PHO(3),PH(3),BOP(3),RC(3),XDP(3),VIP(3)
12 LOGICAL LSURF,LNIT,LSLOPE,LCORNR,LDIF,LDERUG,LTEST
13 COMMON/CEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
14 2,MEP(14),MPX
15 COMMON/EDNAG/VNAG(14,6)
16 COMMON/SORINF/XS(3),VXS(3,3)
17 COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
18 COMMON/LNDFCL/BD(14,6,2)
19 COMMON/THPHW/DT(3),DP(2)
20 COMMON/FIS/PI,TPI,CPR,RPD
21 COMMON/LOGDIF/LSLOPE,LCORNR
22 COMMON/TEST/LDERUG,LTEST
23 COMMON/DOUBLE/IDC(61),ID(14,6),IANG
24 COMMON/HITPLT/IDP
25 COMMON/SURFAC/LSURF(14)
26 FN=FN
27 C!!! INITIALIZE FIELDS
28 EDTH=(0.,0.)
29 EDPH=(0.,0.)
30 ECTH=(0.,0.)
31 ECPH=(0.,0.)
32 IF (LDERUG) WRITE (6,106)
33 106 FORMAT (1,' DEBUGGING DIFPLT SUBROUTINE')
34 C!!! 1. PERFORM DIFFRACTION POINT GEOMETRY CALCULATIONS
35 NC=NC+1
36 IF(NC.GT.MEP(14)) NC=1
37 AFN=PI
38 IF(FKN.GT.2.) AFN=6.-FKN
39 CNP=COS(AFN*PI/2.)
40 SNP=SIN(AFN*PI/2.)
41 DV=0.
42 DO 15 N=1,3
43 15 DV=0.;DO 10 N=1,3
44 10 IF(ABS(DV).GT.1.E-09) GO TO 41
45 DDEL=0.
46 IF (LCORNR) DDEL=0.3
47 DOLN=30*PI*DE,11-DE*1
48 DDM=30*PI*DE,21-DE*1
49 C!!! DETERMINE IF DIFFRACTION EXISTS
50 IF(DV.LT.DOLN.DV.GT.DDM) GO TO 41
51 C!!! COMPUTE EDGE DIFFRACTION POINT
52 CALL DIFFPTS(DV,VI,SP,XI,PI,PI)
53 TH=ATAN2(SQRT(VI(1)**2+VI(2)**2)-VI(1),VI(2))
54 PHIN=ATAN2(VI(2),VI(1))
55 AD=0.
56 VNC=0.
57 DV 15 N=1,3
58 DDPH=0.
59 VNC=VNC+(THIN-ATN(VE,VI(1),VI(2),VI(3))
60 15 ADN=AD+(X(1)-X(14),1,VI(1),VI(2),VI(3))
61 LDIF=THIN
62 C!!! IS DIFFRACTION POINT ON PLATE EDGE?
63 IF (NC.EQ.56) DIFFRACTION POINT AT AN UNPLATE CORNER
64 C!!! AND GET DIFF FIELD
65 IF(VN(14,6,3).EQ.1)

```

```

66      IF(VMG.LT.VMAG(MP,ME)-1.E-4)GO TO 102
67      DO 103 N=1,3
68 103   XDP(N)=X(MP,ME,N)-1.E-4*V(MP,ME,N)
69      LDIF=.FALSE.
70      GO TO 102
71 101   DO 104 N=1,3
72 104   XDP(N)=X(MP,ME,N)+1.E-4*V(MP,ME,N)
73      LDIF=.FALSE.
74 102   DO 16 N=1,3
75      VECT=VP(MP,ME,N)*CNP+VN(MP,N)*SNP
76 16    XDP(N)=XDP(N)+VECT*1.E-5
77 C!!!  2. CHECK TO SEE IF RAY IS SHADOWED
78 C!!!  DETERMINE IF DIFFRACTED RAY HITS ANOTHER PLATE
79      CALL PLAIN(XDP,D,DHIT,MP,LHIT)
80      IF(LHIT)GO TO 42
81 C!!!  DETERMINE IF DIFFRACTED RAY HITS ANOTHER CYLINDER.
82      CALL CYLINT(XDP,D,PHSR,DHIT,LHIT,.TRUE.)
83      IF(LHIT)GO TO 44
84      SPP=0.
85      DO 111 N=1,3
86      VIP(N)=XDP(N)-XS(N)
87 111   SPP=SPP+VIP(N)*VIP(N)
88      SPP=SQRT(SPP)
89      DO 112 N=1,3
90 112   VIP(N)=VIP(N)/SPP
91 C!!!  DOES RAY FROM SOURCE HIT ANOTHER PLATE OR A CYLINDER
92 C!!!  BEFORE DIFFRACTION?
93      CALL PLAIN(XS,VIP,DHIT,MP,LHIT)
94      IF(LHIT.AND.(DHIT.LT.SPP))GO TO 42
95      CALL CYLINT(XS,VIP,PHIR,DHIT,LHIT,.FALSE.)
96      IF(LHIT.AND.(DHIT.LT.SPP))GO TO 44
97      IF (LDEBUG) WRITE (6,*) SP,VI,XD
98      IF (LDEBUG) WRITE (5,*) SPP,VIP,XDP
99 C!!!  3. CALCULATE DIFFRACTION ANGLES AND RELATED GEOMETRY
100     QI=0.
101     PP=0.
102     QD=0.
103     PD=0.
104     DO 20 N=1,3
105     QI=QI-VN(MP,N)*VI(N)
106     PP=PP-VP(MP,ME,N)*VI(N)
107     QD=QD+VN(MP,N)*D(N)
108 20    PD=PD+VP(MP,ME,N)*D(N)
109 C!!!  CALCULATE PSO AND PS, THE INCIDENT AND DIFFRACTED PHI ANGLES
110 C!!!  IN EDGE-FIXED COORDINATE SYSTEM
111     PSOR=BTAN2(QI,PP)
112     PSO=DPR*PSOR
113     IF(PSO.LT.0.) PSO=360.+PSO
114     PSR=BTAN2(QD,PD)
115     PS=DPR*PSR
116     IF(PS.LT.0.) PS=360.+PS
117     PSOD=PSO
118     PSD=PS
119     IF(FN.LE.2.)GO TO 21
120     FN=FN-2.
121     PSOD=360.-PSO
122     PSD=360.-PS
123 21    FNP=FN*180.+1.E-4
124     IF(PSOD.GT.FNP.OR.PSD.GT.FNP)GO TO 41
125 C!!!  IF RAY IS NOT SHADOWED, CHECK TO SEE IF DIFFRACTIONS JUST
126 C!!!  APPEARED. IF SO SET FLAG IN ID(IANG)
127     IF(ID(MP,ME).LE.-1)GO TO 22
128     IDD(IANG)=-(400*ME+20*MP+ID(MP,ME))
129 22    ID(MP,ME)=-2
130     SPHO=SIN(PSOR)
131     CPHO=COS(PSOR)

```

```

132 SPH=SIN(PSK)
133 CPH=COS(PSK)
134 C!!! COMPUTE DIFFRACTION POLARIZATION UNIT VECTORS(PHO,PH,SOP,BO)
135 DO 30 N=1,3
136 PHO(N)=-VP(MP,ME,N)*SPH+VN(MP,N)*CPH
137 30 PH(N)=-VP(MP,ME,N)*SPH+VN(MP,N)*CPH
138 BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
139 BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
140 BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
141 BO(1)=PH(2)*D(3)-PH(3)*D(2)
142 BO(2)=PH(3)*D(1)-PH(1)*D(3)
143 BO(3)=PH(1)*D(2)-PH(2)*D(1)
144 C!!! COMPUTE SBO=SINE(BO)
145 SBO=SQRT((V(MP,ME,3)*D(2)-V(MP,ME,2)*D(3))**2+(V(MP,ME,1)
146 2*D(3)-V(MP,ME,3)*D(1))**2+(V(MP,ME,2)*D(1)-V(MP,ME,1)*D(2))
147 2**2)
148 TPP=SP*SBO*SPO
149 C!!! 4. CALCULATE EDGE DIFFRACTED FIELDS
150 C!!! COMPUTE SOURCE PATTERN FACTORS
151 CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
152 EIPL=EIX*PHO(1)+EIY*PHO(2)+EIZ*PHO(3)
153 EIPL=EIX*BOP(1)+EIY*BOP(2)+EIZ*BOP(3)
154 C!!! IF SLOPE DIFFRACTION IS DESIRED, COMPUTE INCIDENT SLOPE
155 C!!! FIELD PATTERN FACTORS
156 IF(LSLOPE)CALL SOURCE(EIPL,EIPL,VI,PHO,BOP,VXS)
157 C!!! CALCULATE PHASE TERM (REFER PHASE TO RCS ORIGIN)
158 GAM=XD(1)*D(1)+XD(2)*D(2)+XD(3)*D(3)
159 EXPH=CEXP(CMPLX(0.,TPI*(GAM-SP**2))/SORT(SP)
160 C!!! COMPUTE EDGE DIFFRACTION COEFFICIENTS
161 CALL DN(DS,DH,DPS,DPH,TPP,PSD,PSOD,SBO,FI,LSURF(MP))
162 IF(LDEBUG)WRITE(6,*)EIPL,EIPL,EIPL,EIPL
163 IF(LDEBUG)WRITE(6,*)DS,DH,DPS,DPH
164 IF(LDEBUG)WRITE(6,*)TPP,PSD,PSOD,SBO,FI
165 C!!! COMPUTE PERPENDICULAR AND PARALLEL COMPONENTS OF
166 C!!! DIFF. FIELD(EDPR,EDPL)
167 EDPR=-EIPL*DH*EXPH
168 EDPL=-EIPL*DS*EXPH
169 IF(.NOT.LSLOPE)GO TO 201
170 EDPR=EDPR-EIPL*DPH*EXPH/CMPLX(0.,TPI*SP*SBO)
171 EDPL=EDPL-EIPL*DPS*EXPH/CMPLX(0.,TPI*SP*SBO)
172 201 IF(.NOT.LDIF)GO TO 202
173 C!!! COMPUTE THETA AND PHI COMPONENTS OF EDGE DIFF. FIELD.
174 C!!! IF DIFFRACTION EXISTS
175 EDTH=EDPL*(BO(1)*DT(1)+BO(2)*DT(2)+BO(3)*DT(3))
176 2*EDPR*(PH(1)*DT(1)+PH(2)*DT(2)+PH(3)*DT(3))
177 EDPH=EDPL*(BO(1)*DP(1)+BO(2)*DP(2))
178 2*EDPR*(PH(1)*DP(1)+PH(2)*DP(2))
179 C!!! 5. IF CORNER DIFFRACTED FIELD IS DESIRED, CALCULATE
180 C!!! CORNER FIELDS
181 202 IF(.NOT.LCORN)GO TO 40
182 BETN=PSD-PSOD
183 BETP=PSD+PSOD
184 EF=(0.,0.)
185 EG=(0.,0.)
186 MC=MC+1
187 ISN=1
188 C!!! LOOP THRU BOTH CORNERS ON EDGE #ME.
189 35 MC=MC+1
190 IF(MC.GT.MEP(MP))MC=1
191 ISN=-ISN
192 RX=XS(1)-X(MP,MC,1)
193 RY=XS(2)-X(MP,MC,2)
194 RZ=XS(3)-X(MP,MC,3)
195 RM=SQRT(RX*RX+RY*RY+RZ*RZ)
196 CTH=V(MP,ME,1)*RX+V(MP,ME,2)*RY+V(MP,ME,3)*RZ
197 CTH=ISN*CTH/RM

```

```

198      CTHP=ISN*DV
199      THPR=ACOS(CTHP)
200      THR=ACOS(CTH)
201      STHR=SIN(THR)
202      DEL=2.*TPI*RM*(COS(.5*(THR+THPR))**2)
203      ZP=(X(MP,MC,1)-XD(1))*D(1)+(X(MP,MC,2)-XD(2))*D(2)
204      2*(X(MP,MC,3)-XD(3))*D(3)
205      TERM=-STHR/TPI/(CTH+CTHP)/SORT(RM)
206 C!!!  COMPUTE CORNER DIFFRACTION COEFFICIENT(CORN).
207      CORN=-TERM*FCT(DEL)*CEXP(CMPLX(0.,-TPI*(RM-SP-ZP)-.25*PI))
208      CALL DI(ECBI,TPP,BETN,SBO,FN,DEL,.TRUE.)
209      IF(LSURF(MP))GO TO 311
210      CALL DI(ECBR,TPP,BETP,SBO,FN,DEL,.TRUE.)
211 C!!!  COMPUTE MODIFIED EDGE DIFF COEFFICIENTS(DH,DS).
212      DH=ECBI+ECBR
213      DS=ECBI-ECBR
214      GO TO 312
215 311    DH=ECBI
216      DS=(0.,0.)
217 C!!!  COMPUTE CORNER DIFFRACTED FIELD COMPONENTS
218 C!!!  PARALLEL AND PERPENDICULAR TO EDGE
219 312    EDPR=-EIPR*DH*EXPH
220      EDPL=-EIPL*DS*EXPH
221      IF(.NOT.LSLOPE)GO TO 203
222      EDPR=EDPR-EIPR*DPH*EXPH/CMPLX(0.,TPI*SP*SBO)
223      EDPL=EDPL-EIPL*DPS*EXPH/CMPLX(0.,TPI*SP*SBO)
224 C!!!  COMPUTE THETA AND PHI COMPONENTS OF CORNER
225 C!!!  DIFFRACTED FIELDS IN RCS
226 203    ECTH=EDPL*(BO(1)*DT(1)+BO(2)*DT(2)+BO(3)*DT(3))
227      2*EDPR*(PH(1)*DT(1)+PH(2)*DT(2)+PH(3)*DT(3))
228      ECPH=EDPL*(BO(1)*DP(1)+BO(2)*DP(2))
229      2*EDPR*(PH(1)*DP(1)+PH(2)*DP(2))
230 C!!!  COMPUTE TOTAL THETA AND PHI COMPONENTS OF CORNER
231 C!!!  DIFFRACTED FIELDS
232      EF=EF+ECTH*CORN
233      EG=EG+ECPH*CORN
234      IF (.NOT.LDEBUG) GO TO 36
235      WRITE (6,*) DS,DH,EDPR,EDPL
236      WRITE (6,*) ECTH,ECPH,CORN
237      WRITE (6,*) EF,EG
238 36      CONTINUE
239      IF(MC.EQ.ME) GO TO 35
240      ECTH=EF
241      ECPH=EG
242      GO TO 40
243 41      ID(MP,ME)=-1
244      GO TO 40
245 C!!!  IF RAY IS SHADOWED,CHECK TO SEE IF DIFFRACTION
246 C!!!  JUST DISAPPEARED. IF SO SET FLAG IN IDD
247 44      MPH=0
248 42      IF(ID(MP,ME).GE.-1)GO TO 43
249      IDD(IANG)=-((400*ME+20*MP+MPH)
250 43      ID(MP,ME)=MPH
251 40      IF (.NOT.LTEST) GO TO 204
252      WRITE (6,205)
253 205      FORMAT (/, ' TESTING DIFPLT SUBROUTINE' )
254      WRITE (6,*) EDTH,EDPH,ECTH,ECPH
255      WRITE (6,*) FN,ME,MP
256 204      RETURN
257      END

```

DPI

PURPOSE

To calculate the incident part or the reflection part of the wedge slope diffraction coefficient

METHOD

This subroutine computes either the incident part or reflection part of the slope diffraction coefficient based on the uniform Geometrical Theory of Diffraction [10]. This coefficient is given as

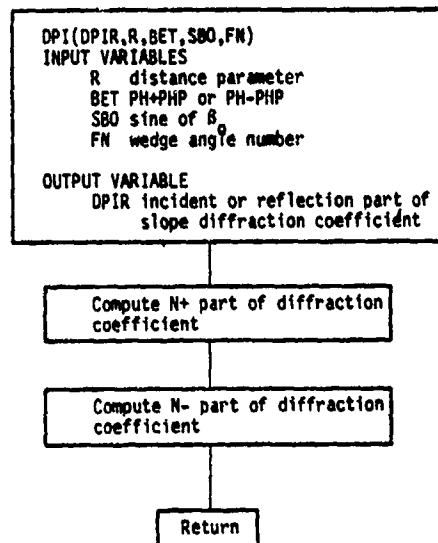
$$\text{DPI}(R, \beta, \sin \beta_0, n) = \frac{-e^{-j \pi/4}}{4n^2 \sqrt{2\pi k} \sin \beta_0} \left\{ \csc^2 \left(\frac{\pi + \beta}{2n} \right) F_s[kRa^+(\beta)] - \csc^2 \left(\frac{\pi - \beta}{2n} \right) F_s[kRa^-(\beta)] \right\},$$

where

$$F_s(x) = 2jx[1 - F(x)]$$

and where β , $a(\beta)$, $F_s(x)$, n are defined in the write up for subroutine DI. An illustration^s of the geometry is given in Figure 55.

FLOW DIAGRAM



SYMBOL DICTIONARY

A	ANGULAR FUNCTION FOR TRANSITION FUNCTION
ANG	BET IN RADIANS
BOTL	ARGUMENT OF TRANSITION FUNCTION
C	REAL PART OF FRESNEL INTEGRAL
COM	CONSTANT FOR SLOPE DIFFRACTION COEFFICIENT
CSCA	COSECANT TIMES THE A FUNCTION
DEK	$8\pi \cdot FN \cdot FN \cdot \sin(BO)$
DN	INTEGER WHICH MOST NEARLY SATISFIES THE EQUATION, $2\pi \cdot FN \cdot DN - BET = \pi$ OR $-\pi$
DNS	COMPUTATIONAL VARIABLE
EX	$\exp(J \cdot K \cdot R \cdot A)$
FPA	SLOPE TRANSITION FUNCTION WITHOUT THE A FUNCTION
N	COMPUTATIONAL VARIABLE
RAG	ARGUMENT OF COSECANT TERM
S	IMAGINARY PART OF FRESNEL INTEGRAL
SON	SIGN OF DNS
TS	$1/\sin^2$
TSIN	SINE OF ARGUMENT OF COSECANT TERM
UNPI	N- COMPONENT OF DPI
UPPI	N+ COMPONENT OF DPI

CODE LISTING

```

1 C-----
2 SUBROUTINE DPI(DPI,R,BET,SRO,FN)
3 C!!!
4 C!!! INCIDENT (BET=PH-PHP) OR REFLECTED (BET=PH+PHP)
5 C!!! PART OF WEDGE SLOPE DIFFRACTION COEFFICIENT
6 C!!!
7 LOGICAL LDEBUG,LTEST
8 COMMON/TEST/LDEBUG,LTEST
9 COMPLEX TOP,COM,EX,UPPI,UNPI,FPA,DPIR
10 COMMON/TOPD/TOP
11 COMMON/PIS/PI,TPI,DPR,RPD
12 IF (LDEBUG) WRITE (6,11)
13 11 FORMAT (/,' DEBUGGING DPI SUBROUTINE')
14 ANG=BET*RPD
15 DEM=4.*TPI*FN*FN*SRO
16 COM=-TOP/DEM
17 C!!! N= PART OF SLOPE DIFFRACTION COEFFICIENT
18 DNS=(PI+ANG)/(2.*FN*PI)
19 SGN=SIGN(1.,DNS)
20 N=IFIX(ABS(DNS)+0.5)
21 DN=SGN*FLOAT(N)
22 A=ABS(1.+COS(ANG-2.*FN*PI*DN))
23 BOTL = 2.*SQRT(ABS(R*A))
24 EX=CEXP(CMPLX(0.,TPI*R*A))
25 CALL FFMELS (C,S,BOTL)
26 C=SQRT(PI/2.)*((0.5-C)
27 S= SQRT(PI/2.)*(S-0.5)
28 FPA=TPI*R*(CMPLX(0.,2.)+4.*SQRT(ABS(TPI*R*A))*EX*CMPLX(C,S))
29 RAG=(PI+ANG)/(2.*FN)
30 TSIN=SIN(RAG)
31 TS=TSIN*TSIN
32 IF(TS.GT.1.E-5) GO TO 442
33 CSCA=-2.*FN*FN*COS(ANG-TPI*FN*DN)/COS((PI+ANG)/FN)
34 GO TO 443
35 442 CSCA=A/TS
36 443 UPPI=COM*CSCA*FPA
37 IF (LDEBUG) WRITE (6,*) DN,A,FPA,UPPI
38 C!!! N= PART OF SLOPE DIFFRACTION COEFFICIENT
39 DNS=(-PI+ANG)/(2.*FN*PI)
40 SGN=SIGN(1.,DNS)
41 N=IFIX(ABS(DNS)+0.5)
42 DN=SGN*FLOAT(N)
43 A=ABS(1.+COS(ANG-2.*FN*PI*DN))
44 BOTL = 2.*SQRT(ABS(R*A))
45 EX=CEXP(CMPLX(0.,TPI*R*A))
46 CALL FFMELS (C,S,BOTL)
47 C=SQRT(PI/2.)*((0.5-C)
48 S= SQRT(PI/2.)*(S-0.5)
49 FPA=TPI*R*(CMPLX(0.,2.)+4.*SQRT(ABS(TPI*R*A))*EX*CMPLX(C,S))
50 RAG=(PI-ANG)/(2.*FN)
51 TSIN=SIN(RAG)
52 TS=TSIN*TSIN
53 IF(TS.GT.1.E-5) GO TO 542
54 CSCA=-2.*FN*FN*COS(ANG-TPI*FN*DN)/COS((PI-ANG)/FN)
55 GO TO 443
56 542 CSCA=A/TS
57 543 UNPI=COM*CSCA*FPA
58 IF (LDEBUG) WRITE (6,*) DN,A,FPA,UNPI
59 DPIR=UPPI-UNPI
60 IF (.NOT.LTEST) GO TO 2
61 WRITE (6,1)
62 1 FORMAT (/,' TESTING DPI SUBROUTINE')
63 WRITE (6,*) DPI,R,BET
64 WRITE (6,*) SRO,FN
65 2 CONTINUE
66 END

```

DPLRCL

PURPOSE

To compute the far-zone electric field for a source ray diffracted off of a given edge on a given plate and then reflected by the cylinder.

PERTINENT GEOMETRY

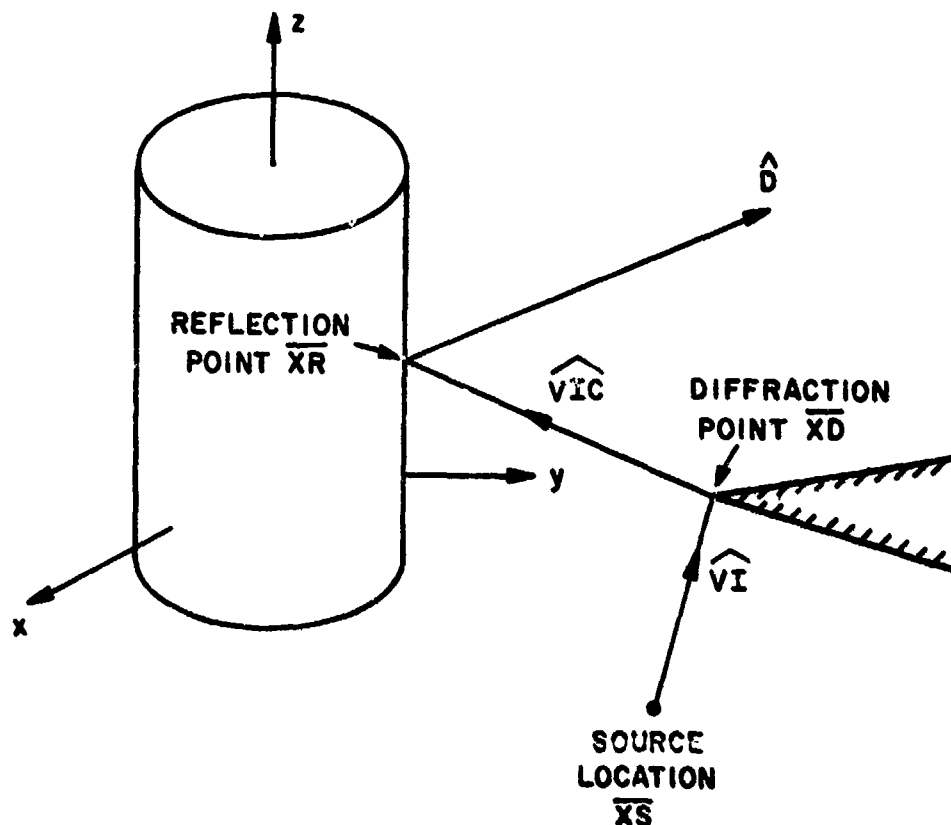


Figure 57--Illustration of a ray diffracted off of a plate edge and then reflected by the cylinder.

METHOD

The field diffracted by a plate edge and then reflected by the elliptic cylinder is calculated in this subroutine. The field diffracted by a plate edge is found using the uniform Geometrical Theory of Diffraction[4]. This causes an astigmatic tube of rays to be incident on the cylinder. The field reflected by the cylinder is found using geometrical optics[4]. The resultant field in the far zone has the form (pp. 163-164, Reference 1)

$$\bar{E}^{d,r} = \bar{E}^i(Q_E) \cdot \bar{D} \cdot \bar{R} \sqrt{\frac{s'}{s''(s'+s'')}} \sqrt{\frac{r}{\rho_1 \rho_2}} e^{-jks''} \frac{e^{-jks}}{s} ,$$

where $\bar{E}^i(Q_E)$ is the incident field on the edge at Q_E , $\bar{D}$ is the dyadic diffraction coefficient, $\bar{R}$ is the dyadic reflection coefficient, ρ_1 and ρ_2 are the reflected ray caustic distances, s' is the distance from the source to the diffraction point, s'' is the distance from the diffraction point to the reflection point, and s is the distance from the reflection point into the far zone. The geometry is shown in Figure 57, and further illustrations can be found in the write ups for subroutines REFCYL and DIFPLT. The phase of the field is referred to the reference coordinate system origin so that

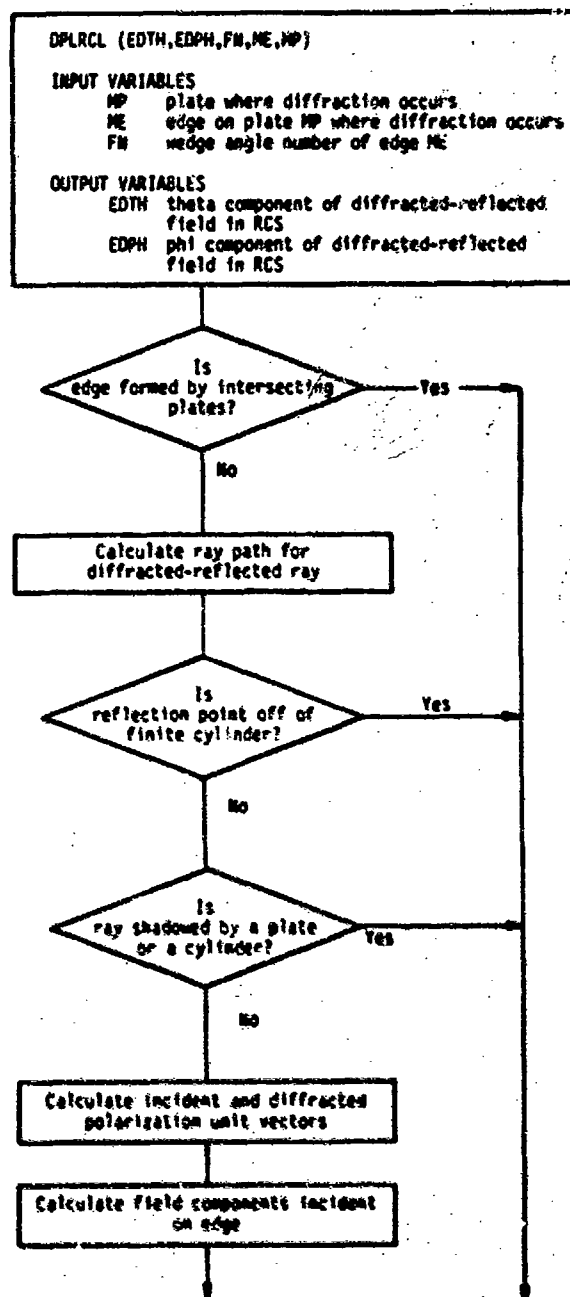
$$\frac{e^{-jks}}{s} = e^{jk\hat{D} \cdot \bar{r}} \frac{e^{-jkR}}{R} .$$

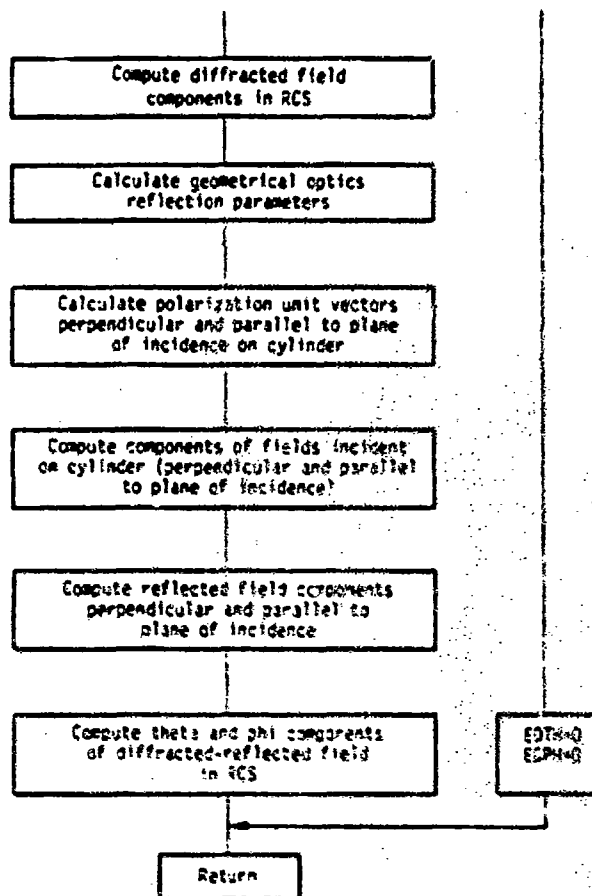
The diffracted-reflected field then has the form

$$\bar{E}^{d,r} = W_m (EDTH\hat{\theta} + EDPH\hat{\phi}) \frac{e^{-jkR}}{R} ,$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM





SYMBOL DICTIONARY

BO	DIFFRACTED FIELD POLARIZATION UNIT VECTOR PARALLEL TO EDGE
BOP	INCIDENT FIELD POLARIZATION UNIT VECTOR PARALLEL TO EDGE
DO1	DOT PRODUCT OF SOURCE RAY DIF FROM PLATE TANGENT TO TAN POINT 1 OF CYLINDER AND PROPAGATION DIRECTION (2-D)
DO2	DOT PRODUCT OF SOURCE RAY DIF FROM PLATE TANGENT TO TAN POINT 2 OF CYLINDER AND PROPAGATION DIRECTION (2-D)
DH	DIFFRACTION COEF. FOR HARD BOUNDARY CONDITION
UHIT	DISTANCE TO HIT POINT ON PLATE
UOTF	TEST VARIABLE USED TO DETERMINE IF REFLECTION IS COMPUTED PROPERLY
DS	DIFFRACTION COEF. FOR SOFT BOUNDARY CONDITION
DV	DOT PRODUCT OF INCIDENT RAY PROPAGATION VECTOR AND EDGE UNIT VECTOR
EDPH	PHI COMPONENT OF EDGE DIFFRACTED REFLECTED E-FIELD
EDPL	COMPONENT OF DIFFRACTED FIELD PARALLEL TO THE EDGE
EDPH	COMPONENT OF DIFFRACTED FIELD PERPENDICULAR TO THE EDGE
EDTH	THETA COMPONENT OF EDGE DIFFRACTED REFLECTED E FIELD
EIPL	COMPONENT OF INCIDENT FIELD PARALLEL TO THE EDGE ON PLANE OF INCIDENCE
EIPN	COMPONENT OF INCIDENT FIELD PERPENDICULAR TO THE EDGE ON PLANE OF INCIDENCE
EIX	SOURCE PATTERN FACTORS FOR X,Y. AND Z COMPONENTS OF INCIDENT E-FIELD
EIY	
EIZ	
ENPP	COMPONENT OF REFLECTED E FIELD PARALLEL TO PLANE OF INCIDENCE
ENPH	COMPONENT OF REFLECTED E FIELD PERPENDICULAR TO PLANE OF INC.
ENX	X,Y,Z COMPONENTS OF REFLECTED FIELD IN RCS
ENY	
ENZ	
EXPH	COMPLEX PHASE AND SPREADING FACTOR
LDAC	SET TRUE IF STARTING POINT INFORMATION EXISTS FROM PREVIOUS PATTERN ANGLE
LHIT	SET TRUE IF PLATE IS HIT
ME	EDGE ON PLATE MP WHERE DIFFRACTION OCCURS
MP	PLATE FOR WHICH DIFFRACTION OCCURS
PU	DOT PRODUCT OF EDGE BINORMAL AND PROPAGATION DIRECTION
PH	DIFFRACTED FIELD PHI UNIT VECTOR PERPENDICULAR TO EDGE
PHIN	PHI COMPONENT OF PROPAGATION DIRECTION OF RAY INCIDENT ON PLATE MP
PHG	INCIDENT FIELD PHI UNIT VECTOR PERPENDICULAR TO EDGE
PP	NEGATIVE DOT PRODUCT OF EDGE BINORMAL AND INCIDENT RAY UNIT VECTOR
PS	DIFFRACTED RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM
PSCH	PHI COMPONENT OF INCIDENT RAY DIRECTION IN EDGE FIXED COORDINATE SYSTEM
PSH	PHI COMPONENT OF DIF RAY PROPAGATION DIRECTION IN EDGE-FIXED COORD SYSTEM
QD	DOT PRODUCT OF PLATE NORMAL AND DIF RAY PROPAGATION DIRECTION
QI	NEGATIVE OF DOT PRODUCT OF PLATE NORMAL AND INCIDENT RAY UNIT VECTOR
QNI1	RADIUS OF CURVATURE PERPENDICULAR TO EDGE OF DIFFRACTED RAY INCIDENT ON REFLECTION POINT
QNI2	RADIUS OF CURVATURE IN EDGE PLANE OF DIFFRACTED RAY INCIDENT ON REFLECTION POINT
WNL1	RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE AT REFLECTION POINT
WNL2	RAY SPREADING RADIUS IN PLANE NORMAL TO PLANE OF INCIDENCE AT CYLINDER REFLECTION POINT
SKAU	DISTANCE FROM DIF POINT TO REFL POINT
SP	DISTANCE FROM SOURCE TO DIFFRACTION POINT (FROM SDR. DEFPPPT)
THIN	THETA COMPONENT OF PROPAGATION DIRECTION OF RAY INCIDENT ON PLATE MP
UB	X,Y COMPONENTS OF UNIT VECTOR TANGENT TO CYL AT REFLECTION POINT

UIPPA } X,Y,Z COMPONENTS OF INCIDENT FIELD POLARIZATION UNIT VECTOR
 UIPPY } PARALLEL TO PLANE OF INCIDENCE
 UIPPZ }
 UIPPX }
 UIPNY } X,Y,Z COMPONENTS OF INC/REFL FIELD POLARIZATION UNIT VECTOR
 UIPNZ } PERPENDICULAR TO PLANE OF INCIDENCE
 UIPNX }
 UN } X,Y COMPONENTS OF UNIT VECTOR NORMAL TO CYL AT REFLECTION POINT
 UAPPA } X,Y,Z COMPONENTS OF REFLECTED FIELD POLARIZATION UNIT VECTOR
 UAPPY } PARALLEL TO PLANE OF INCIDENCE
 UAPPZ }
 VI } UNIT VECTOR OF PROP. DIR OF RAY INCIDENT AT DIFFRACTION POINT (FROM SUB. DEFECT)
 VIC } X,Y,Z COMPONENTS OF UNIT VECTOR OF RAY DIRECTION BETWEEN DIFFRACTION AND REFLECTION
 VU } ELLIPTICAL ANGLE DEFINING REFL. POINT ON CYL (2-0) IN DEGS
 VXS } 3x3 MATRIX DEFINING THE SOURCE COORDINATE SYSTEM AXES
 AX } X,Y,Z COMPONENTS OF DIFFRACTION POINT
 ADP } MODIFIED DIFFRACTION POINT
 AN } X,Y,Z COMPONENTS OF REFLECTION POINT
 AS } SOURCE LOCATION IN REF COORD SYS

CODE LISTING

```

1 C-----
2 SUBROUTINE DPLRC(EDTH,EDPH,FI,ME,MP)
3 C!!
4 C!!! COMPUTES THE FIELD DIFFRACTED FROM EDGE #ME OF PLATE #MP
5 C!!! THEN REFLECTED FROM THE ELLIPTIC CYLINDER
6 C!!!
7 COMPLEX EF,EG,EIPR,EIPL,EXPH,ES,EH,DPS,DPH,EDPR,EDPL,EDTH,EDPH
8 COMPLEX ERPR,ERPP,EIX,EIY,EIZ,ERX,ERY,ERZ
9 DIMENSION UB(2),UB(2),VIC(3),XR(3)
10 DIMENSION VI(3),XD(3),PHO(3),PH(3),BO(3),BO(3),XDP(3)
11 LOGICAL LHIT,LORC,LDEBUG,LTEST
12 COMMON/CEOPL//X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
13 2,MEP(14),MPX
14 COMMON/LOHIF/XS(3),VXS(3,3)
15 COMMON/LIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
16 COMMON/CEONEI/A,B,ZC(2),SIC(2),CNC(2),CTC(2)
17 COMMON/LIDFCL/BO(14,6,2)
18 COMMON/LIDFCL/VDC(14,6),UDC(2),PDCR(14,6,2),TDCR(14,6,2)
19 2,DTDC(14,6),BTDC(14,6,4),BDC(14,6,2)
20 COMMON/LIDFUV/DT(3),DP(2)
21 COMMON/PIS/PI,TPI,DPR,RP)
22 COMMON/TEST/LDEBUG,LTEST
23 COMMON/CLORC/LORC(14,6)
24 C!!! IS EDGE FORMED BY INTERSECTING PLATES?
25 IF(FI.GT.2.) GO TO 40
26 C!!! IS DIFFRACTION POSSIBLE?
27 IF(D(3).GT.DXC(MP,ME,1).OR.D(3).LT.DDC(MP,ME,2)) GO TO 39
28 PD1=BTDC(MP,ME,1)*CPHS+BTDC(MP,ME,2)*SPHS
29 PD2=BTDC(MP,ME,3)*CPHS+BTDC(MP,ME,4)*SPHS
30 IF(D(3).GT.DXC(MP,ME).AND.PD2.GT.DTDC(MP,ME)) GO TO 39
31 C!!! CALCULATE RAY PATH FOR DIFFRACTED-REFLECTED FIELD
32 CALL DLEPT(VR,XR,DOTP,DD,SHAG,VIC,XD,SP,VI,DV,ME,MP
33 2,LORC(MP,ME))
34 IF(DOTP.LE.0.) GO TO 40
35 IF(DV.LT.DDC(MP,ME,1).OR.DV.GT.DDC(MP,ME,2)) GO TO 40
36 C!!! IS REFLECTION OF POINT OFF OF FINITE CYLINDER?
37 IF(XR(3).GT.ZC(1)+XR(1)*CTC(1).OR.
38 2,XR(3).LT.ZC(2)+XR(1)*CTC(2)) GO TO 40
39 CND=COS(PH+0.5*PI)
40 SNP=SIN(PH+0.5*PI)
41 DO 10 N=1,3
42 VECT=VDC(MP,ME,N)*CND+VN(MP,N)*SNP
43 XDP(N)=XD(N)+VECT*1.E-5
44 C!!! IS RAY SHADOWED BY A PLATE OR A CYLINDER?
45 CALL PLINT(MP,VIC,DHIT,MP,LHIT)
46 IF(LHIT.AND.(DHIT.LT.SHAG)) GO TO 40
47 CALL PLINT(AS,VI,DHIT,MP,LHIT)
48 IF(LHIT.AND.(DHIT.LT.SP)) GO TO 40
49 CALL PLINT(AS,D,UNIT,0,UNIT)
50 IF(LHIT) GO TO 40
51 THIR=ATAN2(SORT(VI(1)+VI(1)+VI(2)+VI(2)),VI(3))
52 PHIR=DT/AZ(VI(2),VI(1))
53 CALL CYLINT(AS,VI,PHIR,DHIT,UNIT,.FALSE.)
54 IF(LHIT.AND.(DHIT.LT.SP)) GO TO 40
55 CEND.
56 DD=0.
57 DD=0.
58 DD=0.
59 DO 20 N=1,3
60 DI=DI-VDC(MP,ME,VI(N))
61 DP=DP-VDC(MP,ME,VI(N))
62 DC=DC-VDC(MP,ME,VI(N))
63 PL=PL-VDC(MP,ME,VI(N))
64 PSOR=ATAN2(DI,DI)
65 PSOR=PSOR
66 IF(PSO.LT.0.) PSO=2*PI+PSO

```

```

67 PSR=PI/2(PI,PD)
68 PS=1.0*PSR
69 IF(PS,1,1,0.) PS=360./PS
70 FNP=PI*180./PI-4
71 IF(PS,GT.FNP.OR.PS,GT.FNP) GO TO 40
72 SPHO=SIN(PSOR)
73 CPHO=COS(PSOR)
74 SPH=SIN(PSR)
75 CPH=COS(PSR)
76 C!!! CALCULATE INCIDENT AND DIFFRACTED POLARIZATION
77 C!!! UNIT VECTORS
78 DO 30 I=1,3
79 PHO(I)=-VP(MP,ME,N)*SPHO+VH(MP,N)*CPHO
80 30 PH(I)=-VP(MP,ME,N)*SPH+VH(MP,N)*CPH
81 BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
82 BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
83 BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
84 BO(1)=PH(2)*VIC(3)-PH(3)*VIC(2)
85 BO(2)=PH(3)*VIC(1)-PH(1)*VIC(3)
86 BO(3)=PH(1)*VIC(2)-PH(2)*VIC(1)
87 C!!! CALCULATE SOURCE FIELD PATTERN FACTOR
88 CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PIIR,VXS)
89 EIPR=EIX*PHO(1)+EIY*PHO(2)+EIZ*PHO(3)
90 EIPL=EIX*BOP(1)+EIY*BOP(2)+EIZ*BOP(3)
91 SFC=SQRT((V(MP,ME,3)*VIC(2)-V(MP,ME,2)*VIC(3))**2
92 + (V(MP,ME,1)*VIC(3)-V(MP,ME,3)*VIC(1))**2
93 + (V(MP,ME,2)*VIC(1)-V(MP,ME,1)*VIC(2))**2)
94 TPP=SP*SMAG*SEO*SEO/(SP+SMAG)
95 EXPH=CEXP(CMPLX(C.,-TPI*SP))/SQRT(SP)
96 C!!! CALCULATE DIFFRACTED FIELDS
97 CALL DR(DS,DI,DPS,DPH,TPP,PS,PSO,SEO,PI,.,FALSE.)
98 EDPR=-EIPR*DI*EXPH
99 EDPL=-EIPL*DS*EXPH
100 C!!! CALCULATE GEOMETRICAL OPTICS REFLECTION
101 C!!! PARAMETERS
102 RG=LD*DU*DD/A/B
103 CALL NADIR(U1,UB,VR)
104 CTHC=UH(1)*D(1)+UH(2)*D(2)
105 RH=RTAN2(-VIC(1)*UR(1)-VIC(2)*UH(2),-VIC(3))
106 RHI1=SMAG
107 RHI2=SMAG+SP
108 TH1=PH(1)*UH(1)+PH(2)*UB(2)
109 TH2=PH(3)
110 TH21=BO(1)*UB(1)+BO(2)*UB(2)
111 TH22=BO(3)
112 DET=TH11*TH22-TH12*TH21
113 CTHD=CTHC/(DET*DET)
114 RHA=.5*(1./RHI1+1./RHI2)+CTHD*(TH22*TH22+TH12*TH12)/RG
115 RHI2=1./RHI1-1./RHI2
116 RHB=RHI2*RHI2+RHI2*4.*CTHD*(TH22*TH22+TH12*TH12)/RG
117 RHB=RHB+4.*CTHD*CTHD*((TH22*TH22+TH12*TH12)/RG)**2
118 RHE=.5*SQRT(RHB)
119 RHO1=1./(RHA+RHE)
120 RHO2=1./(RHA-RHE)
121 C!!! COMPUTE POLARIZATION UNIT VECTORS (PERPENDICULAR
122 C!!! AND PARALLEL TO PLANE OF INCIDENCE)
123 UIPRX=SIN(RH-.5*PI)*UH(1)
124 UIPRY=SIN(RH-.5*PI)*UB(2)
125 UIPRZ=COS(RH-.5*PI)
126 UIPRX=VIC(3)*UIPRY-VIC(2)*UIPRZ
127 UIPRY=VIC(1)*UIPRZ-VIC(3)*UIPRX
128 UIPRZ=VIC(2)*UIPRX-VIC(1)*UIPRY
129 URPE=D(3)*UIPRY-D(2)*UIPRZ
130 URPR=D(1)*UIPRZ-D(3)*UIPRX
131 URPRZ=D(2)*UIPRX-D(1)*UIPRY
132 EXPH=CEXP(CMPLX(C.,-TPI*SMAG))/SQRT(SMAG*(SP+SMAG))

```

```

133 C!!! CALCULATE DIFFRACTED FIELD COMPONENTS INCIDENT
134 C!!! ON CYLINDER PARALLEL AND PERP. TO PLANE OF INC.
135 EIPR=EDPL*(BO(1)*UIPRX+BO(2)*UIPRY+BO(3)*UIPRZ)
136 2+EDPR*(PH(1)*UIPRX+PH(2)*UIPRY+PH(3)*UIPRZ)
137 EIPL=EDPL*(BO(1)*UIPPX+BO(2)*UIPPY+BO(3)*UIPPZ)
138 2+EDPR*(PH(1)*UIPPX+PH(2)*UIPPY+PH(3)*UIPPZ)
139 C!!! COMPUTE REFLECTED FIELD COMPONENTS PARALLEL
140 C!!! AND PERPENDICULAR TO CYLINDER
141 ERPR=-SQRT(RHO1*RHO2)*EXPH*EIPR
142 ERPP=SQRT(RHO1*RHO2)*EXPH*EIPL
143 C!!! CALCULATE X,Y,Z COMPONENTS OF REFLECTED FIELD
144 ERX=ERPR*UIPRX+ERPP*URPPX
145 ERY=ERPR*UIPRY+ERPP*URPPY
146 ERZ=ERPR*UIPRZ+ERPP*URPPZ
147 EXPH=CEXP(CMPLX(0.,TPI*(XR(1)*D(1)+XR(2)*D(2)+XR(3)*D(3)))
148 C!!! COMPUTE THETA AND PHI COMPONENTS OF DIFFRACTED-
149 C!!! REFLECTED FIELD IN RCS
150 EDTH=(ERX*DT(1)+ERY*DT(2)+ERZ*DT(3))*EXPH
151 EDPH=(ERX*DP(1)+ERY*DP(2))*EXPH
152 GO TO 940
153 39 LDRC(MP,ME)=.FALSE.
154 40 CONTINUE
155 EDTH=(0.,0.)
156 EDPH=(0.,0.)
157 500 CONTINUE
158 IF(.NOT.LTEST) RETURN
159 WRITE(6,901)
160 501 FORMAT(/,' TESTING DPLRCL SUBROUTINE')
161 WRITE(6,*) EDTH,EDPH,FN,ME,MP
162 RETURN
163 END

```

DPLRPL

PURPOSE

To calculate the far-zone electric field (with phase referred to the RCS origin) for a source ray which diffracts off of edge ME of plate MP and is then reflected by plate MR.

PERTINENT GEOMETRY

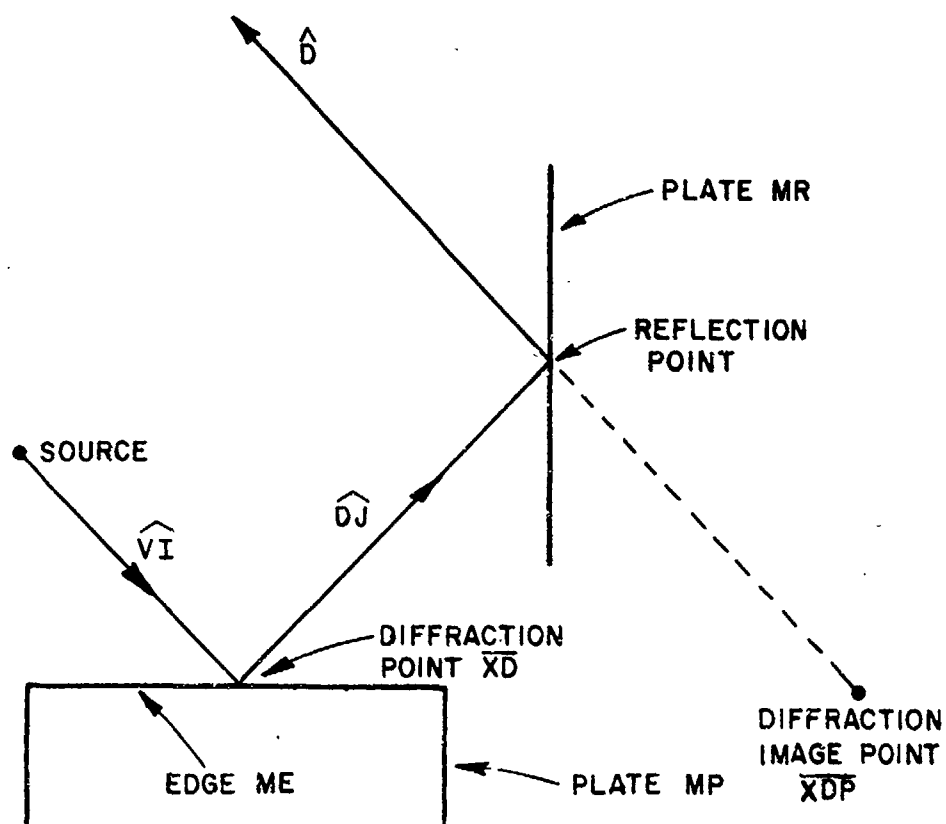


Figure 58--Illustration of edge-diffracted, plate-reflected ray.

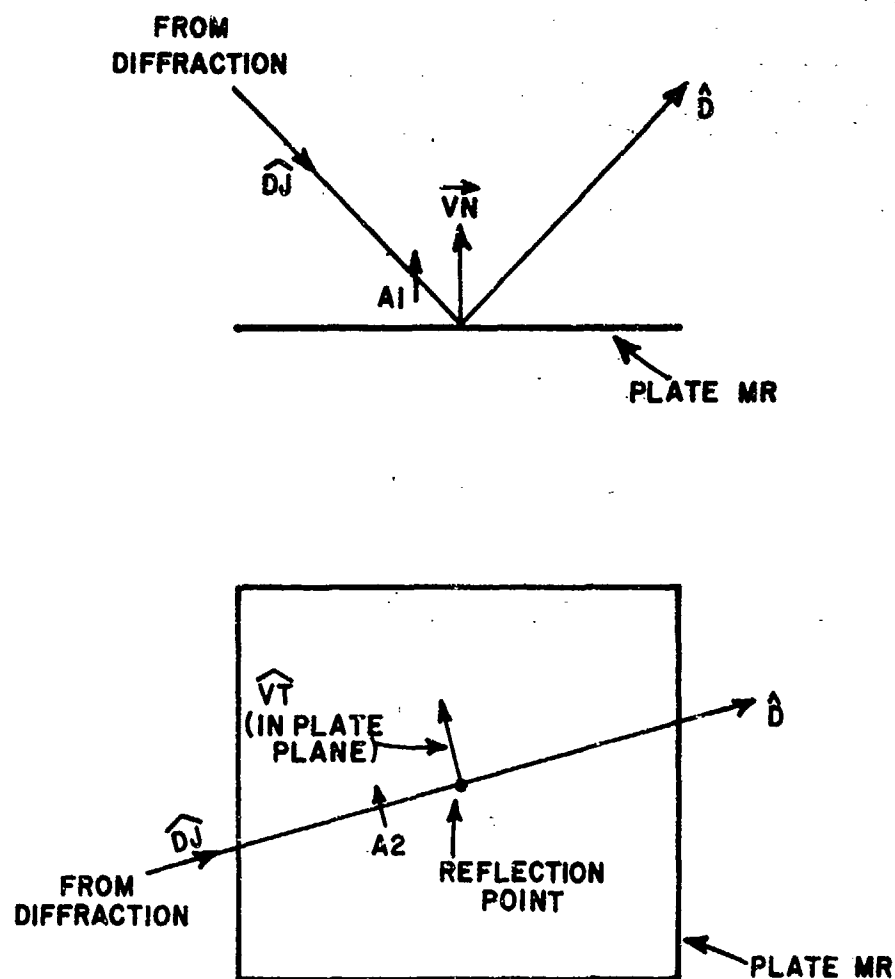


Figure 59--Geometry used in computing plate reflection.

METHOD

The fields diffracted by a plate edge and the reflected by another plate are calculated in this subroutine[4,9,10]. The diffracted and slope diffracted fields of the plate edges and corners are obtained as described in subroutine DIFPLT. The reflection from the plate is found by decomposing the diffracted fields into components tangent and normal to the reflection plate (see Figure 59), satisfying the appropriate boundary conditions and then transforming the field back to the reference coordinate system. The edge and slope diffracted fields are combined and the phase referred to the reference coordinate system origin by the factor $e^{jk\hat{D} \cdot \vec{XDP}}$. The form of the

the field is therefore given by

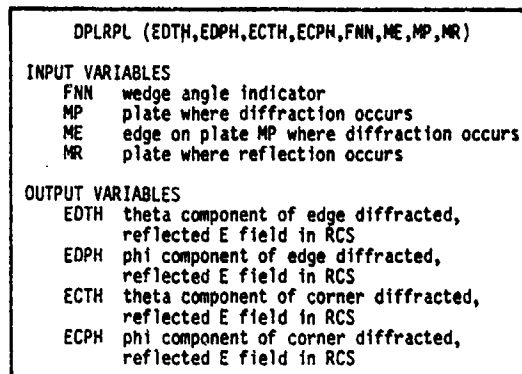
$$\bar{E}^d = W_m (EDTH\hat{\theta} + EDPH\hat{\phi}) \frac{e^{-jkR}}{R} .$$

The corner diffracted and slope corner diffracted fields are combined in a similar way and are given by

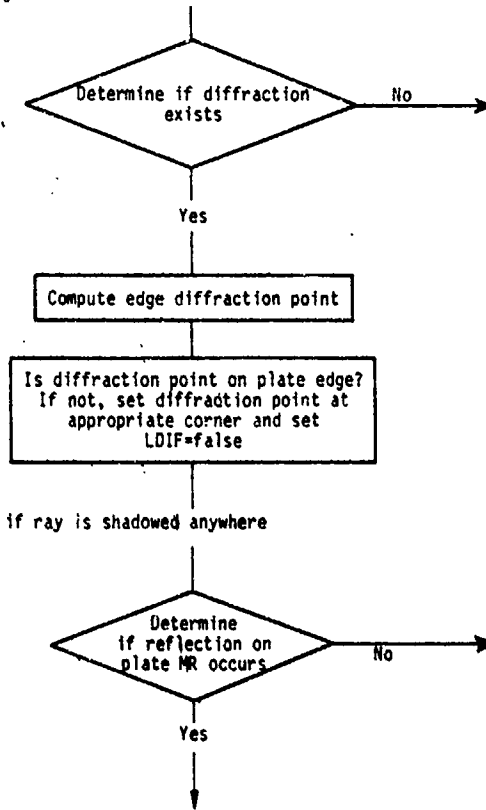
$$\bar{E}^c = W_m (ECTH\hat{\theta} + ECPH\hat{\phi}) \frac{e^{-jkR}}{R} ,$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

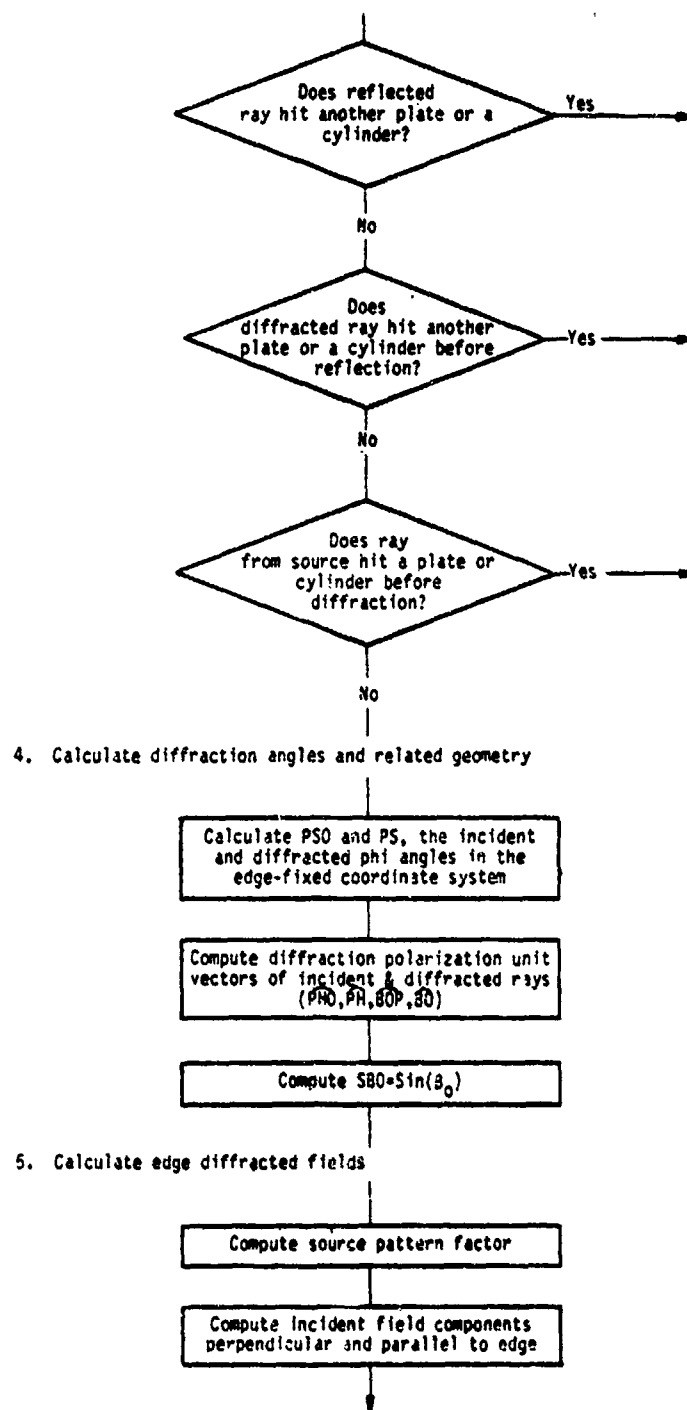
FLOW DIAGRAM

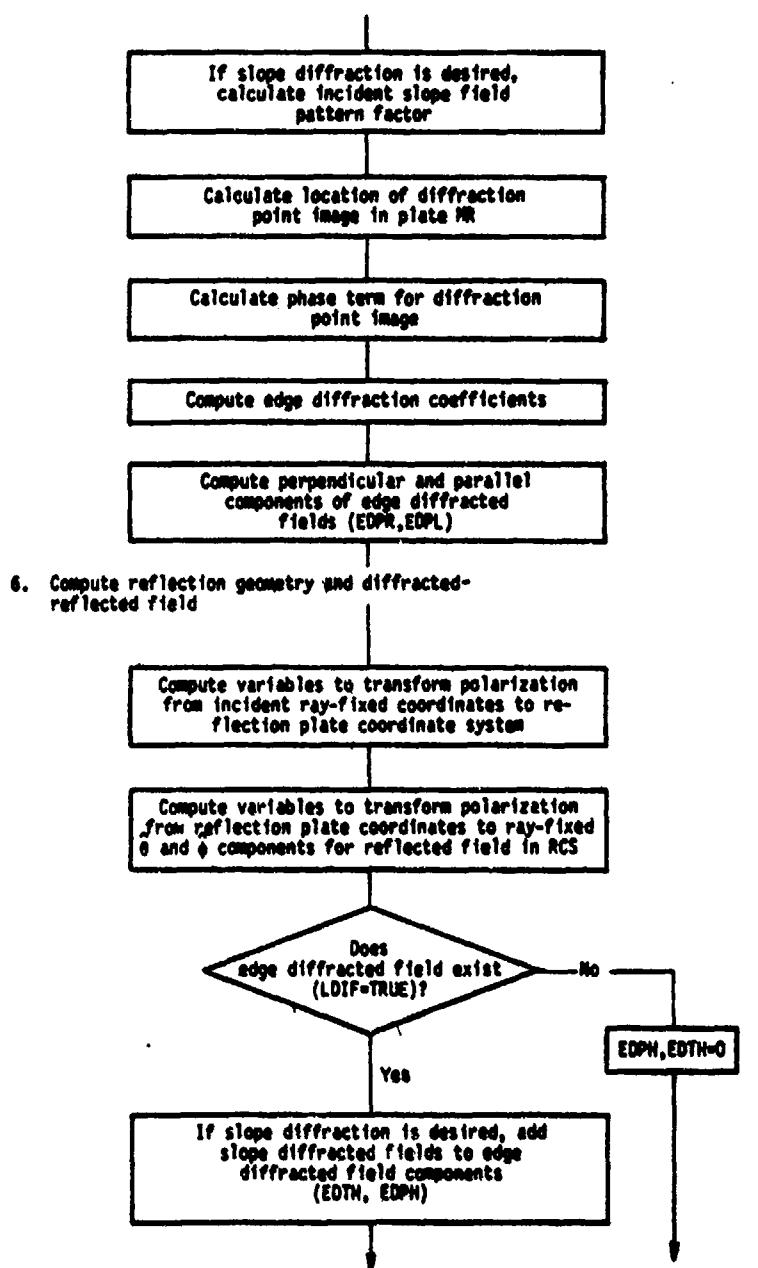


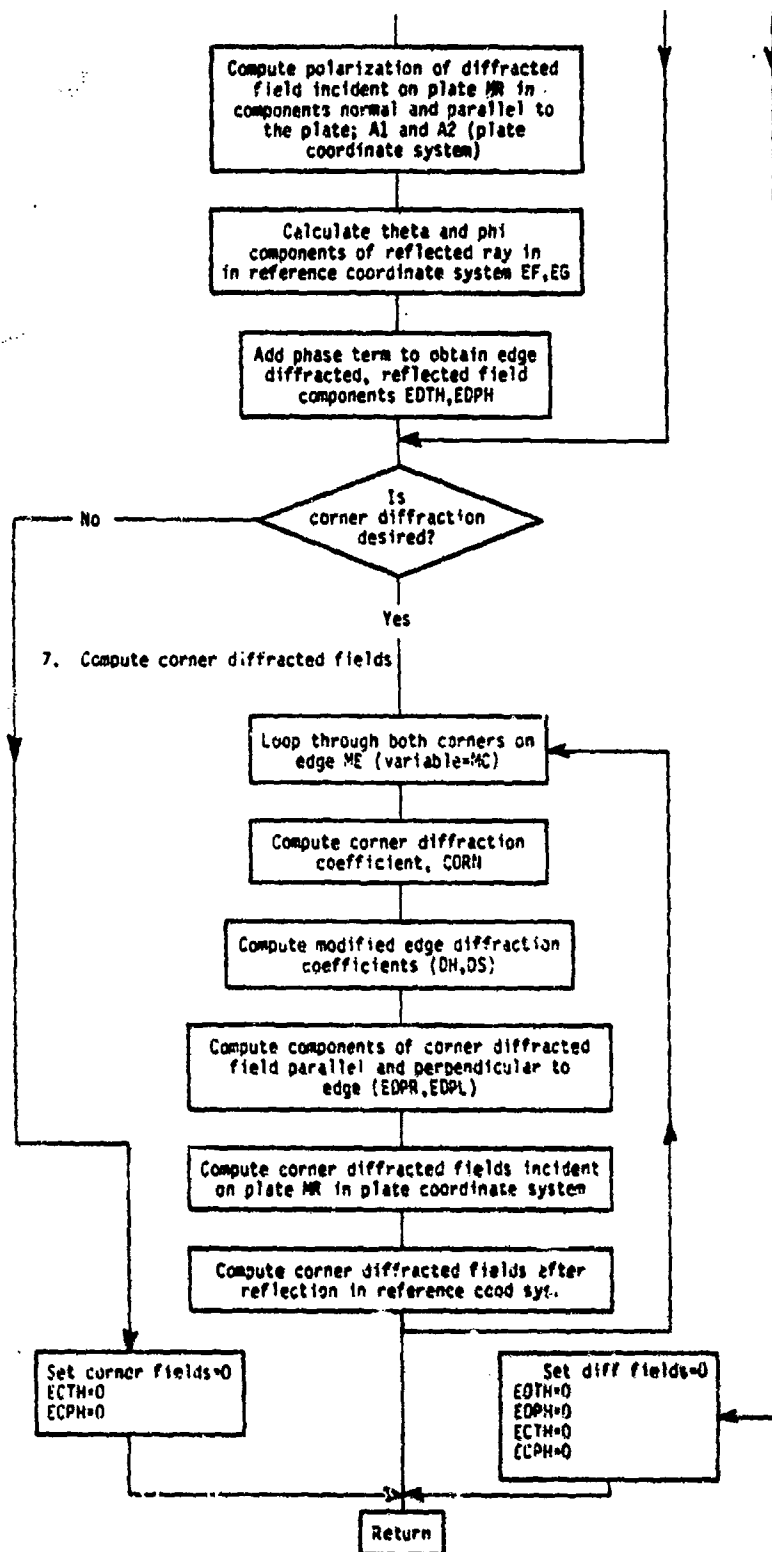
1. Compute direction $\hat{D}_j$ of ray incident on plate MR (ray propagation direction after diffraction).
2. Perform diffraction point geometry calculations to obtain diffracted ray in direction $\hat{D}_j$



3. Check to see if ray is shadowed anywhere







SYMBOL DICTIONARY

A1	COMPONENT OF INCIDENT DIF FIELD NORMAL TO PLATE MR
A2	COMPONENT OF INCIDENT DIF FIELD TANGENT TO PLATE MR
A3	DETERMINANT OF TRANSFORMATION MATRIX
ADN	DOT PRODUCT OF VECTOR FROM PLATE MP TO THE SOURCE AND THE PLATE UNIT NORMAL
AFN	WEDGE ANGLE NUMBER
BDEL	VARIABLE USED TO EXPAND DIFFRACTION ANGLE RANGE IF CORNER DIFFRACTION IS USED
BDHI	UPPER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA
BDLOW	LOWER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA
BEIN	DIFFERENCE IN DIFFRACTED AND INCIDENT PHI ANGLES
BETP	SUM OF DIFFRACTED AND INCIDENT PHI ANGLES
BO	DIFFRACTED FIELD BETA POLARIZATION UNIT VECTOR (IN EDGE FIXED COORD SYS) IN RCS COMPONENTS (FOR DIF EDGE) IN (X,Y,Z) REF COORD SYS. COMPONENTS
BOP	INCIDENT FIELD BETA POLARIZATION UNIT VECTOR (IN EDGE FIXED COORD SYS) IN RCS COMPONENTS (FOR DIF EDGE)
C11	DOT PRODUCT OF REFLECTED FIELD POLARIZATION VECTOR DT AND PLATE COORD SYS UNIT VECTOR VN
C11A	DOT PRODUCT OF RAY-FIXED C.S. VECTOR BO AND PLATE C.S. VECTOR VN
C12	DOT PRODUCT OF RAY FIXED COORD SYS VECTOR DP AND PLATE COORD SYS UNIT VECTOR VN
C12A	DOT PRODUCT OF RAY-FIXED C.S. VECTOR PH AND PLATE C.S. VECTOR VT
C21	DOT PRODUCT OF RAY FIXED COORD SYS VECTOR DT AND PLATE COORD SYS UNIT VECTOR VT
C21A	DOT PRODUCT OF RAY FIXED COORD SYS VECTOR BO AND PLATE COORD SYS VECTOR VT
C22	DOT PROD. OF REFLECTED FIELD POLARIZATION UNIT VECTOR DP AND PLATE COORD SYS UNIT VECTOR VT
C22A	DOT PRODUCT OF RAY-FIXED C.S. VECTOR PH AND PLATE C.S. VECTOR VT
CNP	COSINE OF HALF WEDGE ANGLE
CORN	CORNER DIFFRACTION COEFFICIENT
CPH	COSINE OF PSR
CPHJ	COSINE OF PHJR
CPHO	COSINE OF PSOR
CTH	COSINE OF THR
CTHJ	COSINE OF THJR
CTHP	COSINE OF THPR
DEL	PARAMETER USED IN TRANSITION FUNCTION
DH	DIFFRACTION COEF. FOR HARD BOUNDARY CONDITION
DHIT	DISTANCE FROM SOURCE TO NEAREST HIT (FROM SUBS. PLAIN OR CYLINT)
DHT	DISTANCE FROM SOURCE TO HIT (RETURNED FROM PLAIN AND CYLINT)
DJ	X,Y, AND Z COMPONENTS OF RAY PROP. DIRECTION BETWEEN DIFFRACTION AND REFLECTION
DPH	SLOPE DIFFRACTION COEFFICIENT FOR HARD BOUNDARY CONDITION
DPS	SLOPE DIFFRACTION COEFFICIENT FOR SOFT BOUNDARY CONDITION
DS	DIFFRACTION COEF. FOR SOFT BOUNDARY CONDITION
DV	DOT PRODUCT OF EDGE VECTOR AND DIFFRACTED RAY PROPAGATION DIRECTION UNIT VECTOR, DJ
ECBI	DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR INCIDENT DIFFRACTED FIELD, MODIFIED FOR CORNER DIFFRACTION
ECBH	EDGE DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR REFLECTED DIFFRACTED FIELD, MODIFIED FOR CORNER DIFFRACTION
ECPH	PHI COMPONENT OF CORNER DIFFRACTED, REFLECTED E-FIELD
ECTH	THETA COMPONENT OF CORNER DIFFRACTED, REFLECTED E-FIELD
EDPH	PHI COMPONENT OF EDGE DIFFRACTED, REFLECTED E-FIELD
EUPL	COMPONENT OF DIFFRACTED FIELD PARALLEL TO THE EDGE
EDPH	COMPONENT OF DIFFRACTED FIELD PERPENDICULAR TO THE EDGE
EDTH	THETA COMPONENT OF EDGE DIFFRACTED, REFLECTED E-FIELD
EF	THETA COMPONENT OF PATTERN FACTOR OF FIELD INCIDENT ON EDGE ALSO THETA COMPONENT OF REFLECTED FIELD
EG	PHI COMPONENT OF PATTERN FACTOR OF FIELD INCIDENT ON EDGE ALSO PHI COMPONENT OF REFLECTED FIELD IN RCS
EIPL	COMPONENT OF INCIDENT FIELD PARALLEL TO THE EDGE

EIPLP PATTERN FACTOR FOR COMPONENT OF SOURCE (INCIDENT) SLOPE FIELD
 PARALLEL TO THE EDGE (RAY INCIDENT ON DIFF EDGE)
 EIPR COMPONENT OF INCIDENT FIELD PERPENDICULAR TO THE EDGE
 EIPRP PATTERN FACTOR FOR COMPONENT OF SOURCE (INCIDENT) SLOPE FIELD
 PERPENDICULAR TO THE EDGE (RAY INCIDENT ON DIFF EDGE)
 EIX } SOURCE PATTERN FACTORS FOR X,Y, AND Z COMPONENTS OF INCIDENT
 EIZ } FIELD ON EDGE
 EXPH COMPLEX PHASE TERM (REFER PHASE TO RCS. ORIGIN)
 FN WEDGE ANGLE NUMBER
 FNN WEDGE ANGLE INDICATOR
 FNP ANGLE EXTERIOR TO WEDGE IN DEGREES
 GAM DOT PRODUCT OF THE PROPAGATION DIRECTION AND THE VECTOR FROM
 THE REF COORD SYS ORIGIN TO THE DIFFRACTION POINT IMAGE LOCATION
 ISN SIGN CHANGE VARIABLE
 LHIT SET TRUE IF RAY HITS A PLATE OR CYLINDER (FROM PLAIN OR CYLIND)
 MC CORNER AT END OF EDGE ME
 ME EDGE ON PLATE MP WHERE DIFFRACTION OCCURS
 MP PLATE FOR WHICH DIFFRACTION OCCURS
 MR PLATE WHERE REFLECTION OCCURS
 N DO LOOP VARIABLE
 PD DOT PRODUCT OF EDGE BINORMAL AND PROPAGATION DIRECTION
 PH DIFFRACTED FIELD PHI POLARIZATION UNIT VECTOR (IN EDGE
 FIXED COORDINATE SYSTEM) IN RCS COMPONENTS (FOR DIF EDGE)
 PHIR PHI COMPONENT OF INCIDENT RAY DIRECTION IN REF COORD SYS.
 PHJR PHI COMPONENT OF RAY PROP. DIR. BETWEEN DIF AND REFLECTION
 IN RCS
 PHO INCIDENT FIELD PHI POLARIZATION UNIT VECTOR (IN EDGE
 FIXED COORD SYS) IN RCS COMPONENTS (FOR DIF EDGE)
 PHSR PHI COMPONENT OF PROPAGATION DIRECTION AFTER REFL IN RCS
 PP NEGATIVE DOT PRODUCT OF EDGE BINORMAL AND INCIDENT RAY UNIT NORMA
 PS PSR*DPR
 PSD DIFFRACTED RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM
 PSO PSR*DPR
 PSOD INCIDENT RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM
 PSOR PHI COMPONENT OF INCIDENT RAY DIRECTION IN EDGE
 FIXED COORDINATE SYSTEM
 PSR PHI COMPONENT OF DIF RAY DIRECTION IN EDGE-FIXED COORD SYS
 QD DOT PRODUCT OF PLATE NORMAL AND PROPAGATION DIRECTION
 QI NEGATIVE OF DOT PRODUCT OF PLATE NORMAL AND INCIDENT RAY
 PROPAGATION DIRECTION
 NM MAGNITUDE OF VECTOR FROM CORNER MC TO SOURCE
 NX } X,Y, AND Z COMPONENTS OF VECTOR FROM CORNER MC TO SOURCE
 NY }
 NZ }
 SBO SINE OF BO, THE ANGLE THE DIFFRACTED RAY MAKES WITH THE EDGE
 SNP SINE OF HALF WEDGE ANGLE
 SP DISTANCE FROM SOURCE TO DIFFRACTION POINT (FROM SUB. DFPTND)
 SPH SINE OF PSR
 SPHJ SINE OF PHJR
 SPHO SINE OF PSOR
 SPP DISTANCE FROM SOURCE TO MODIFIED DIFFRACTION POINT
 STHJ SINE OF THJH
 STHR SINE OF THR
 TERM COEFFICIENT OF CORNER DIFFRACTED FIELDS
 THIR THETA COMPONENT OF INCIDENT RAY DIRECTION IN REF COORD SYS
 THJR THETA COMPONENT OF RAY PROP. DIR. BETWEEN DIF. AND REFLECTION
 IN RCS
 THPH ANGLE DIFFRACTED RAY MAKES WITH EDGE
 THN ANGLE BETWEEN EDGE UNIT VECTOR AND RAY FROM SOURCE
 TO CORNER MC
 TPP DISTANCE PARAMETER USED IN CALCULATING DIFFRACTION COEFFICIENTS
 VECT VECTOR USED TO MOVE DIFFRACTION POINT OFF EDGE FOR
 SHADOWING TESTS
 VI UNIT VECTOR OF RAY INCIDENT ON EDGE FROM SOURCE
 (FROM SUBROUTINE DFPTND)

VIP	UNIT VECTOR FROM SOURCE TO MODIFIED DIFFRACTION POINT
VMO	DISTANCE ALONG THE EDGE FROM FIRST CORNER OF EDGE ME TO DIFFRACTION POINT
VT	X, Y, AND Z COMPONENTS OF UNIT VECTOR ON PLATE MR NORMAL TO PLANE OF INCIDENCE (TANGENT TO PLATE)
VXS	3X3 MATRIX DEFINING THE SOURCE COORDINATE SYSTEM AXES
XD	DIFFRACTION POINT (CALCULATED IN SUB. DFPTWD)
XDP	MODIFIED DIFFRACTION POINT USED FOR SHADOWING TESTS
	ALSO, LOCATION OF DIFF POINT IMAGE IN PLATE MR
XDPP	DIFFRACTION POINT, CONVERTED TO REFLECTION HIT POINT
XS	SOURCE LOCATION IN REF COORD SYS
ZP	DOT PRODUCT OF PROPAGATION UNIT VECTOR AND VECTOR FROM DIFFRACTION POINT TO CORNER MC

CODE LISTING

```

1 C-----
2 SUBROUTINE DPLRPL(EDTH,EDPH,ECTH,ECPH,FNN,ME,NP,MR)
3 C!!!
4 C!!! DETERMINES THE DIFFRACTED/REFLECTED FIELD WITH PHASE
5 C!!! REFERRED TO ORIGIN. RAY IS DIFFRACTED FROM EDGE #ME ON
6 C!!! PLATE #NP AND REFLECTED FROM PLATE #MR
7 C!!!
8 COMPLEX EF,EG,EIPR,EIPL,EXPH,DS,DH,DPS,DPH,EDPR,EDPL,EDTH,EDPH
9 COMPLEX EIPRP,EIPLP,EIX,EIY,EIZ,CORN,FFCT,A1,A2
10 COMPLEX ECTH,ECPH,ECBI,ECBR
11 DIMENSION VI(3),XD(3),PHO(3),PH(3),BOP(3),BO(3),XDP(3)
12 DIMENSION DJ(3),VT(3),VIP(3),XDP(3)
13 LOGICAL LSUMF,LHIT,LDEBUG,LTEST,LSLOPE,LCORNR,LDIF
14 COMMON/TEST/LDEBUG,LTEST
15 COMMON/LOGDIF/LSLOPE,LCORNR
16 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
17 2,MEP(14),MPX
18 COMMON/SORINF/XS(3),VXS(3,3)
19 COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
20 COMMON/BNDFCL/BD(14,6,2)
21 COMMON/THPHUV/DT(3),DP(2)
22 COMMON/PIS/PI,TP1,DPR,RPD
23 COMMON/EDMAG/VNAG(14,6)
24 COMMON/SURFAC/LSURF(14)
25 FNN=FNN
26 MC=ME+1
27 IF (MC.GT.MEP(NP)) MC=1
28 C!!! INITIALIZE FIELDS
29 EDTH=(0.,0.)
30 EDPH=(0.,0.)
31 ECTH=(0.,0.)
32 ECPH=(0.,0.)
33 C!!! 1. COMPUTE INCIDENT DIRECTION OF FIELD ON PLATE #MR
34 CALL REFBP(PHJR,THJR,PHSR,THSR,MR)
35 SPHJ=SIN(PHJR)
36 CPHJ=COS(PHJR)
37 STHJ=SIN(THJR)
38 CTHJ=COS(THJR)
39 DJ(1)=CPHJ*STHJ
40 DJ(2)=SPHJ*STHJ
41 DJ(3)=CTHJ
42 C!!! 2. PERFORM DIFFRACTION POINT GEOMETRY CALCULATIONS
43 C!!! TO OBTAIN RAY IN DIRECTION DJ
44 DV=0.
45 DO 10 N=1,3
46 10 DV=DV+DJ(N)*V(NP,ME,N)
47 IF (ABS(DV).GT.0.999) GO TO 49
48 BOEL=0.
49 IF (LCORNR) BOEL=0.3
50 BOLOW=BD(NP,ME,1)+BOEL
51 BDHI=BD(NP,ME,2)+BOEL
52 C!!! DETERMINE IF DIFFRACTION EXISTS
53 IF (DV.LT.BOLOW.OR.DV.GT.BDHI) GO TO 49
54 C!!! COMPUTE EDGE DIFFRACTION PT.
55 CALL DEFTWD(XS,DV,VI,SP,XD,ME,NP)
56 ADN=0.
57 VNG=0.
58 AFN=FNN
59 IF (AFN.GT.2.)AFN=0.-AFN
60 CNP=COS(AFN*PI/2.)
61 SNP=SIN(AFN*PI/2.)
62 DO 15 N=1,3
63 XDP(N)=XD(N)
64 VNG=VNG+(XD(N)-X(NP,ME,N))*V(NP,ME,N)
65 15 ADN=ADN+(XS(N)-X(NP,1,N))*VN(NP,N)
66 LDIF=.TRUE.

```

```

67 C111 IS DIF POINT ON PLATE EDGE? IF NOT SET DIF POINT AT
68 C111 APPROPRIATE CORNER AND SET LDIF FALSE
69 IF (VMG.LT.1.E-5) GO TO 101
70 IF (VMG.LT.VMAG(MP,ME)-1.E-4) GO TO 102
71 DO 103 N=1,3
72 103 XDP(N)=X(MP,ME,N)-1.E-4*V(MP,ME,N)
73 LDIF=.FALSE.
74 GO TO 102
75 101 DO 104 N=1,3
76 104 XDP(N)=X(MP,ME,N)+1.E-4*V(MP,ME,N)
77 LDIF=.FALSE.
78 102 DO 106 N=1,3
79 VECT=VP(MP,ME,N)*CNP+VN(MP,N)*SNP
80 XDP(N)=XDP(N)+1.E-5*VECT
81 10 XDP(N)=XDP(N)
82 C111 3. CHECK TO SEE IF RAY IS SHADOWED ANYWHERE
83 C111 DETERMINE IF REFLECTION OFF PLATE #MR OCCURS
84 CALL PLAIN(XDP,DJ,DHT,-MR,LHIT)
85 IF(.NOT.LHIT) GO TO 40
86 C111 DETERMINE IF RAY AFTER REFLECTION HITS PLATE
87 CALL PLAIN(XDP,D,DHT,MR,LHIT)
88 IF(LHIT) GO TO 40
89 C111 DETERMINE IF RAY AFTER REFLECTION HITS CYLINDER
90 CALL CYLINT(XDP,D,PHSR,DHT,LHIT,.TRUE.)
91 IF(LHIT) GO TO 40
92 C111 DETERMINE IF EDGE DIF. RAY HITS PLATE BEFORE REFLECTION
93 CALL PLAIN(XDP,DJ,DHT,MR,LHIT)
94 IF(LHIT.AND.(DHT.LT.DHT)) GO TO 40
95 C111 DETERMINE IF EDGE DIF. RAY HITS CYLINDER BEFORE REFLECTION
96 CALL CYLINT(XDP,DJ,PHJR,DHT,LHIT,.TRUE.)
97 IF(LHIT.AND.(DHT.LT.DHT)) GO TO 40
98 SPP=0.
99 DO 111 N=1,3
100 VIP(N)=XDP(N)-XS(N)
101 111 SPP=SPP+VIP(N)*VIP(N)
102 SPP=SQRT(SPP)
103 DO 112 N=1,3
104 112 VIP(N)=VIP(N)/SPP
105 C111 DETERMINE IF RAY FROM SOURCE HITS A PLATE OR A CYLINDER
106 C111 BEFORE DIF.
107 CALL PLAIN(XS,VIP,DHT,MP,LHIT)
108 IF(LHIT.AND.(DHT.LT.SPP)) GO TO 40
109 THIR=ATAN2(SQRT(VI(1)*VI(1)+VI(2)*VI(2)),VI(3))
110 PHIR=ATAN2(VI(2),VI(1))
111 CALL CYLINT(XS,VI,PHIR,DHT,LHIT,.FALSE.)
112 IF(LHIT.AND.(DHT.LT.SPP)) GO TO 40
113 C111 4. CALCULATE DIFFRACTION ANGLES AND RELATED GEOMETRY
114 OI=0.
115 PP=0.
116 OD=0.
117 PD=0.
118 DO 20 N=1,3
119 OI=OI-VN(MP,N)*VT(N)
120 PP=PP-VP(MP,ME,N)*VI(N)
121 OD=OD-VN(MP,N)*OJ(N)
122 20 PD=PD-VP(MP,ME,N)*OJ(N)
123 C111 CALCULATE PS0 AND PS, THE INCIDENT AND DIFFRACTED PHI ANGLES
124 C111 IN EDGE-FIXED COORDINATE SYSTEM
125 PS0=ATAN2(OI,PP)
126 PS0=OPR+PS0
127 IF(PS0.LT.0.) PS0=360.+PS0
128 PSR=ATAN2(OD,PD)
129 PS=OPR+PSR
130 IF(PS.LT.0.) PS=360.+PS
131 PS0=PS0
132 PS=PS

```

```

133 IF(FN.LE.2.)GO TO 21
134 FN=FN-2.
135 PSOD=360.-PSO
136 PSD=360.-PS
137 21 FNP=FN*180./1.E-4
138 IF(PSO.GT.FNP.OR.PS.GT.FNP) GO TO 40
139 SPHO=SIN(PSOR)
140 CPHO=COS(PSOR)
141 SPH=SIN(PSR)
142 CPH=COS(PSR)
143 C!!! COMPUTE DIFFRACTION POLARIZATION UNIT VECTORS(PHO,PH,BOP,BO)
144 DO 30 N=1,3
145 PHO(N)=-VP(MP,ME,N)*SPHO+VN(MP,N)*CPHO
146 30 PH(N)=-VP(MP,ME,N)*SPH+VN(MP,N)*CPH
147 BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
148 BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
149 BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
150 BO(1)=PH(2)*DJ(3)-PH(3)*DJ(2)
151 BO(2)=PH(3)*DJ(1)-PH(1)*DJ(3)
152 BO(3)=PH(1)*DJ(2)-PH(2)*DJ(1)
153 C!!! COMPUTE SBO=SINE(BO)
154 SBO=SQRT((V(MP,ME,3)*DJ(2)-V(MP,ME,2)*DJ(3))**2+(V(MP,ME,1)
155 2*DJ(3)-V(MP,ME,3)*DJ(1))**2+(V(MP,ME,2)*DJ(1)-V(MP,ME,1)
156 2*DJ(2))**2)
157 TPP=SP*SBO*SBO
158 C!!! 5. CALCULATE EDGE DIFFRACTED FIELDS
159 C!!! COMPUTE SOURCE PATTERN FACTORS
160 CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
161 C!!! COMPUTE INCIDENT FIELD COMPONENTS PARALLEL AND PERP. TO EDGE
162 EIPL=EIX*PHO(1)+EIY*PHO(2)+EIZ*PHO(3)
163 EIPL=EIX*BOP(1)+EIY*BOP(2)+EIZ*BOP(3)
164 C!!! IF SLOPE DIF IS DESIRED, COMPUTE INCIDENT SLOPE FIELD
165 C!!! PATTERN FACTORS
166 IF(1LSLOPE)CALL SOURCP(EIPLP,EIPLP,VI,PHO,BOP,VXS)
167 C!!! CALCULATE LOCATION OF DIF POINT IMAGE IN PLATE MR
168 CALL IMAGE(XDP,XD,ADN,MR)
169 C!!! CALCULATE PHASE TERM FOR DIF IMAGE POINT
170 GAM=XDP(1)*D(1)+XDP(2)*D(2)+XDP(3)*D(3)
171 EXPR=CEXP(CMPLX(0.,TPI*(GAM-SP)))/SQRT(SP)
172 C!!! COMPUTE EDGE DIFFRACTION COEFFICIENTS
173 CALL DMDS,DH,DPS,DPH,TPP,PSD,PSOD,SBO,FN,LSURF(MP)
174 IF (LDEBUD) WRITE (6,*) EIPL,EIPLP,EIPLP,EIPLP
175 IF (LDEBUD) WRITE (6,*) DS,DH,DPS,DPH
176 IF (LDEBUD) WRITE (6,*) TPP,PSD,PSOD,SBO,FN
177 C!!! COMPUTE PERPENDICULAR AND PARALLEL COMPONENTS OF
178 C!!! EDGE DIFFRACTED FIELD (EDPR,EDPL)
179 EDPR=EIPL*DH
180 EDPL=EIPL*DS
181 C!!! IF SLOPE DIF IS DESIRED, ADD SLOPE FIELDS TO EDGE DIF
182 C!!! FIELD COMPONENTS
183 IF(.NOT.1LSLOPE)GO TO 201
184 EDPR=EDPR-EIPLP*DPH/CMPLX(0.,TPI*(SP+SBO))
185 EDPL=EDPL-EIPLP*DPS/CMPLX(0.,TPI*(SP+SBO))
186 C!!! 6. COMPUTE EDGE DIFFRACTED REFLECTED RAY
187 C!!! COMPUTE VARIABLES TO TRANSFORM POLARIZATION FROM INCIDENT
188 C!!! RAY FIXED COORD SYS TO REFLECTION PLATE COORD SYS
189 201 VT(1)=VN(MR,2)*DJ(3)-VN(MR,3)*DJ(2)
190 VT(2)=VN(MR,3)*DJ(1)-VN(MR,1)*DJ(3)
191 VT(3)=VN(MR,1)*DJ(2)-VN(MR,2)*DJ(1)
192 C11A=VN(MR,1)*BO(1)+VN(MR,2)*BO(2)+VN(MR,3)*BO(3)
193 C12A=VN(MR,1)*PH(1)+VN(MR,2)*PH(2)+VN(MR,3)*PH(3)
194 C21A=VT(1)*BO(1)+VT(2)*BO(2)+VT(3)*BO(3)
195 C22A=VT(1)*PH(1)+VT(2)*PH(2)+VT(3)*PH(3)
196 C!!! COMPUTE VARIABLES TO TRANSFORM RAY POLARIZATION FROM PLATE
197 C!!! COORDINATES TO RAY-FIXED THETA AND PHI COMPONENTS FOR REFL
198 C!!! RAY IN RCS

```

```

199      C11=VN(MR,1)*DT(1)+VN(MR,2)*DT(2)+VN(MR,3)*DT(3)
200      C12=VN(MR,1)*DP(1)+VN(MR,2)*DP(2)
201      C21=VT(1)*DT(1)+VT(2)*DT(2)+VT(3)*DT(3)
202      C22=VT(1)*DP(1)+VT(2)*DP(2)
203      A3=C11*C22-C12*C21
204 C!!! DETERMINE IF EDGE DIF FIELD EXISTS
205      IF (.NOT.LDIF) GO TO 202
206 C!!! COMPUTE POLARIZATION OF DIF FIELD INCIDENT ON PLATE MR
207 C!!! IN COMPONENTS NORMAL AND TANGENT TO THE PLATE (A1 AND A2)
208      A1=EDPL*C11A+EDPR*C12A
209      A2=EDPL*C21A+EDPR*C22A
210 C!!! CALCULATE THETA AND PHI COMPONENTS OF REFL FIELD IN RCS
211      EF=(A1*C22+A2*C12)/A3
212      EG=(A2+C11+A1*C21)/A3
213 C!!! ADD PHASE TERM TO OBTAIN DIF REFL FIELD COMPONENTS
214 C!!! EDTH AND EDPH
215      EDTH=EF*EXPH
216      EDPH=EG*EXPH
217 C!!! 7. IF CORNER DIF FIELD IS DESIRED, COMPUTE CORNER FIELDS
218 202 IF (.NOT.LCORN) GO TO 40
219      BETN=PSD-PSOD
220      BETP=PSD+PSOD
221      MC=ME-1
222      ISN=1
223 C!!! LOOP THRU BOTH CORNERS ON EDGE #ME
224 35 MC=MC+1
225      IF(MC.GT.MP) MC=1
226      ISN=-ISN
227      RX=XS(1)-X(MP,MC,1)
228      RY=XS(2)-X(MP,MC,2)
229      RZ=XS(3)-X(MP,MC,3)
230      RM=SQRT(RX*RX+RY*RY+RZ*RZ)
231      CTH=V(MP,ME,1)*RX+V(MP,ME,2)*RY+V(MP,ME,3)*RZ
232      CTH=ISN*CTH/RM
233      CTHP=ISN*DV
234      THPR=ACOS(CTHP)
235      THR=ACOS(CTH)
236      STHR=SIN(THR)
237      DEL=2.*TPI*RM*(COS(.5*(THR+THPR))**2)
238      ZP=(X(MP,MC,1)-XD(1))*DJ(1)+(X(MP,MC,2)-XD(2))*DJ(2)
239      2+(X(MP,MC,3)-XD(3))*DJ(3)
240      TERM=-STHR/TPI/(CTH+CTHP)/SQRT(RM)
241 C!!! COMPUTE CORNER DIFFRACTION COEFFICIENT (CORN).
242      CORN=TERM*FFCT(DEL)*CEXP(CMPLX(0.,-TPI*(RM-SP-ZP)-.25*PI))
243      CALL DI(ECBI,TPP,BETN,SBO,FN,DEL,.TRUE.)
244      IF(LSURF(MP))GO TO 311
245      CALL DI(ECBR,TPP,BETP,SBO,FN,DEL,.TRUE.)
246 C!!! COMPUTE MODIFIED EDGE DIFF. COEFFICIENTS (DH,DS).
247      DH=ECBI+ECBR
248      DS=ECBI-ECBR
249      GO TO 312
250 311 DH=ECBI
251      DS=(0.,0.)
252 C!!! COMPUTE COMPONENTS OF CORNER DIFFRACTED FIELD PARALLEL
253 C!!! AND PERPENDICULAR TO EDGE
254 312 EDPR=-EIPR*DH*EXPH
255      EDPL=-EIPL*DS*EXPH
256      IF(.NOT.LSLOPE)GO TO 203
257      EDPR=EDPR-EIPR*DPH*EXPH/CMPLX(0.,TPI*SP*SBO)
258      EDPL=EDPL-EIPL*DPS*EXPH/CMPLX(0.,TPI*SP*SBO)
259 C!!! COMPUTE CORNER DIFFRACTED FIELDS INCIDENT ON PLATE MR IN
260 C!!! PLATE COORDINATE SYSTEM
261 203 A1=EDPL*C11A+EDPR*C12A
262      A2=EDPL*C21A+EDPR*C22A
263 C!!! COMPUTE CORNER DIFFRACTED FIELDS AFTER REFLECTION IN RCS
264      EF=(A1*C22+A2*C12)/A3

```

```

265      EG=-(A2*C11+A1*C21)/A3
266 C!!!  COMPUTE THETA AND PHI COMPONENTS OF CORNER DIFFRACTED
267 C!!!  REFLECTED FIELDS (ECTH, ECPH) IN RCS
268      ECTH=ECTH+EF*CORN
269      ECPH=ECPH+EG*CORN
270      IF (.NOT.LDEBUG) GO TO 36
271      WRITE (6,*) DS,DH,EDPR,EDPL
272      WRITE (6,*) ECTH,ECPH,CORN
273      WRITE (6,*) EF,EG
274 36    CONTINUE
275      IF(MC.EQ.ME) GO TO 35
276 40    IF (.NOT.LTEST) GO TO 204
277      WRITE (6,205)
278 205    FORMAT (/, ' TESTING DPLRPL SUBROUTINE' )
279      WRITE (6,*) EDTH,EDPH,ECTH,ECPH
280      WRITE (6,*) FN,ME,MP,MR
281 204    RETURN
282      END

```

DPTNFW

PURPOSE

To compute the diffraction point for a ray which is diffracted by a given edge and observed at a specified near field point of the plate.

PERTINENT GEOMETRY

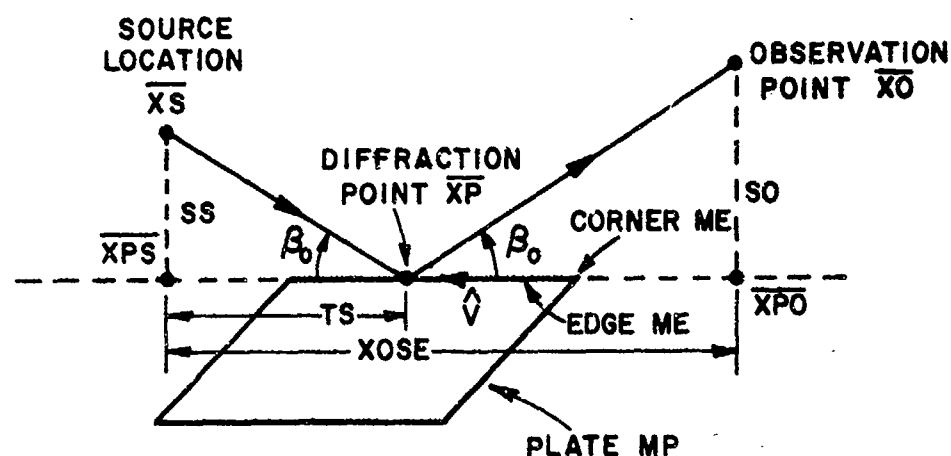


Figure 60-- Geometry for finding the diffraction point with the observation point in the near field of the plate.

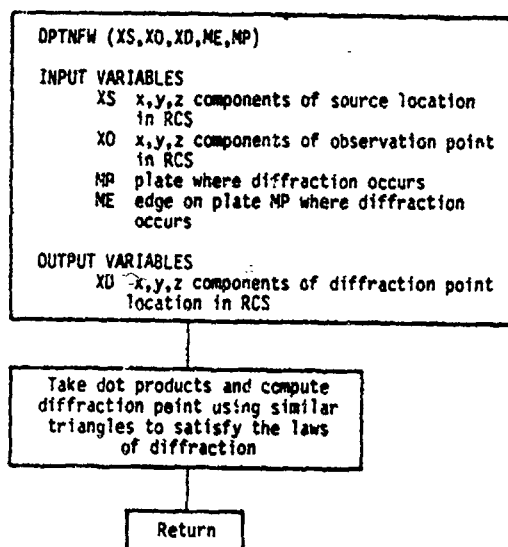
METHOD

The diffraction point is found using similar triangles defined by perpendiculars from the source and observation points to the edge line. The diffraction point is given by

$$XD = XPS + \frac{SS \times XOSE}{SS + SO} \hat{V},$$

where the above quantities are illustrated in Figure 60.

FLOW DIAGRAM



SYMBOL DICTIONARY

SS	DISTANCE FROM SOURCE TO POINT XPS
SO	DISTANCE FROM OBSERVATION POINT TO POINT XPO
IS	DISTANCE FROM XPS TO XD ALONG EDGE LINE
XOPE	DOT PRODUCT OF RAY FROM CORNER ME TO OBSERVATION POINT AND EDGE UNIT VECTOR
XOSE	DOT PRODUCT OF RAY FROM SOURCE TO OBSERVATION POINT AND EDGE UNIT VECTOR
XPS	POINT ON LINE THROUGH EDGE ME CLOSEST TO SOURCE
XPO	POINT ON LINE THROUGH EDGE ME CLOSEST TO OBSERVATION POINT
XSCE	DOT PRODUCT OF RAY FROM CORNER ME TO SOURCE AND EDGE UNIT VECTOR

CODE LISTING

```

1 C-----
2      SUBROUTINE DPTNFW(XS,XO,XD,ME,MP)
3 C!!!
4 C!!! DETERMINES THE NEAR FIELD DIFFRACTION POINT ON A PLATE EDGE
5 C!!!
6      DIMENSION XS(3),XO(3),XPS(3),XPO(3),XD(3)
7      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
8      2,MEP(14),MPX
9      XSCE=0.
10     XOCE=0.
11     XOSE=0.
12     DO 10 N=1,3
13       XSCE=XSCE+(XS(N)-X(MP,ME,N))*V(MP,ME,N)
14       XOCE=XOCE+(XO(N)-X(MP,ME,N))*V(MP,ME,N)
15 10    XOSE=XOSE+(XO(N)-XS(N))*V(MP,ME,N)
16     DO 20 N=1,3
17       XPS(N)=XSCE+V(MP,ME,N)+X(MP,ME,N)
18 20    XPO(N)=XOCE+V(MP,ME,N)+X(MP,ME,N)
19     SS=(XS(1)-XPS(1))*(XS(1)-XPS(1))+(XS(2)-XPS(2))*(XS(2)-XPS(2))
20     2+(XS(3)-XPS(3))*(XS(3)-XPS(3))
21     SS=SQRT(SS)
22     SO=(XO(1)-XPO(1))*(XO(1)-XPO(1))+(XO(2)-XPO(2))*(XO(2)-XPO(2))
23     2+(XO(3)-XPO(3))*(XO(3)-XPO(3))
24     SO=SQRT(SO)
25     TS=SS*XOSE/(SS+SO)
26     DO 30 N=1,3
27 30    XD(N)=XPS(N)+TS*V(MP,ME,N)
28     RETURN
29     END

```

DQG32

PURPOSE

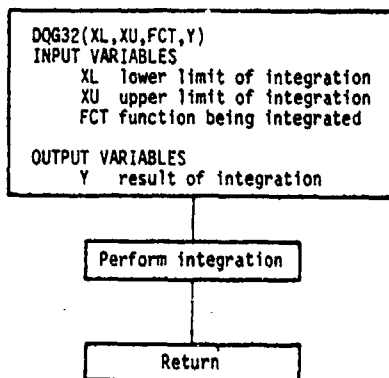
To numerically integrate a given function over a specified range.

METHOD

This subroutine uses a 32 point Gaussian quadrature formula to compute the integral of a function[11]. The form of the integral is given as

$$Y = \int_{XL}^{XU} FCT(x)dx .$$

FLOW DIAGRAM



SYMBOL DICTIONARY

FCT	FUNCTION DEFINING THE INTEGRAND
XL	LOWER BOUND OF INTEGRAL
XU	UPPER BOUND OF INTEGRAL
Y	RESULT OF INTEGRAL

CODE LISTING

```

1 C-----
2      SUBROUTINE DCG32 (XL, XU, FCT, Y)
3 C!!!
4 C!!! 32 POINT GAUSSIAN QUADRATURE INTEGRATION ROUTINE
5 C!!!
6      A=.5D0*(XU+XL)
7      B=XU-XL
8      C=.49863193092474D0*B
9      Y=.55093050047350D-2*(FCT(A+C)+FCT(A-C))
10     C=.49280575577263D0*B
11     Y=Y+.8137197365452D-2*(FCT(A+C)+FCT(A-C))
12     C=.48238112779375D0*B
13     Y=Y+.12696032654631D-1*(FCT(A+C)+FCT(A-C))
14     C=.46745303796886D0*B
15     Y=Y+.17136931456510D-1*(FCT(A+C)+FCT(A-C))
16     C=.44816057788302D0*B
17     Y=Y+.21417949011113D-1*(FCT(A+C)+FCT(A-C))
18     C=.42468380686628D0*B
19     Y=Y+.25499029631188D-1*(FCT(A+C)+FCT(A-C))
20     C=.39724189798397D0*B
21     Y=Y+.29342046739267D-1*(FCT(A+C)+FCT(A-C))
22     C=.36609105937014D0*B
23     Y=Y+.32911111388180D-1*(FCT(A+C)+FCT(A-C))
24     C=.33152213346510D0*B
25     Y=Y+.36172897054424D-1*(FCT(A+C)+FCT(A-C))
26     C=.29385787862038D0*B
27     Y=Y+.39096947893535D-1*(FCT(A+C)+FCT(A-C))
28     C=.25344995446611D0*B
29     Y=Y+.41655962113473D-1*(FCT(A+C)+FCT(A-C))
30     C=.21067563806531D0*B
31     Y=Y+.43826046502201D-1*(FCT(A+C)+FCT(A-C))
32     C=.16593430114106D0*B
33     Y=Y+.45586939347881D-1*(FCT(A+C)+FCT(A-C))
34     C=.11964368112606D0*B
35     Y=Y+.46922199540402D-1*(FCT(A+C)+FCT(A-C))
36     C=.7223598079139D-1*B
37     Y=Y+.47819360039637D-1*(FCT(A+C)+FCT(A-C))
38     C=.24153832843869D-1*B
39     Y=B*(Y+.48270044257363D-1*(FCT(A+C)+FCT(A-C)))
40     RETURN
41     END

```

DW

PURPOSE

To determine wedge and slope diffraction coefficients for the soft and hard boundary conditions.

METHOD

This subroutine calculates the edge diffraction and slope diffraction coefficients for the hard and soft boundary conditions using the Uniform Geometrical Theory of Diffraction[4,5]. The edge diffraction coefficient has the form

$$D_b = DI(R, \phi - \phi', \sin \beta_0, n) \pm DI(R, \phi + \phi', \sin \beta_0, n),$$

where D_b is for the hard case and D_s is for the soft case and n is the wedge angle number (FN).

The slope diffraction coefficient has the form

$$\frac{\partial D_b}{\partial \phi'} = DPI(R, \phi - \phi', \sin \beta_0, n) \mp DPI(R, \phi + \phi', \sin \beta_0, n).$$

In both cases the $\phi - \phi'$ part refers to the incident part of the diffraction coefficient and $\phi + \phi'$ refers to the reflection part. For grazing incidence where $\phi' = 0$, the diffraction coefficients have the form

$$D_h = DI(L, \phi, \sin \beta_0, n)$$

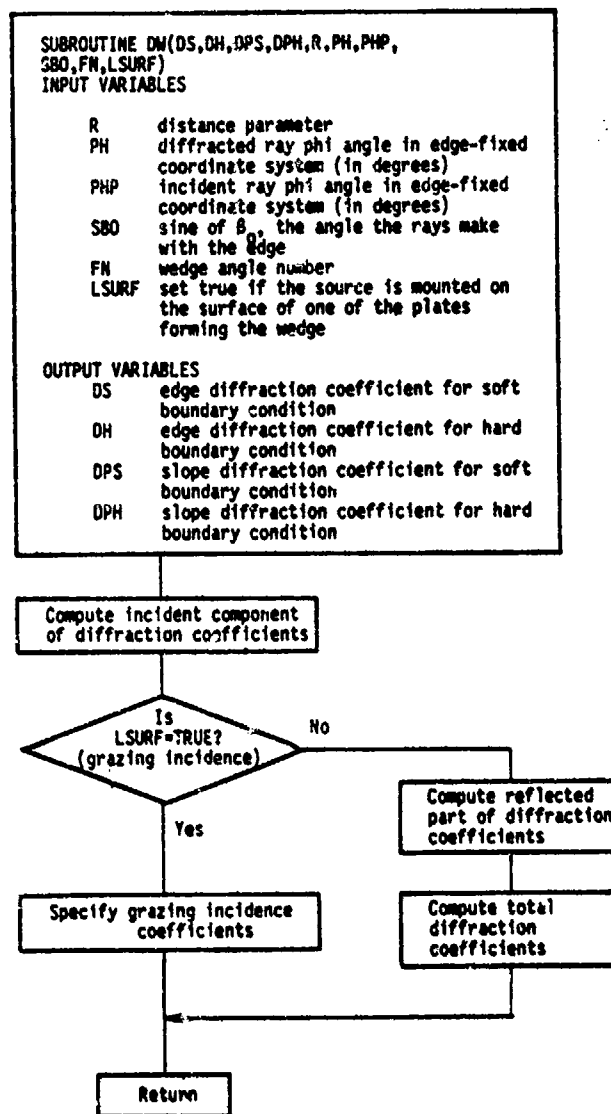
$$D_s = 0$$

$$\frac{\partial D_s}{\partial \phi'} = DPI(L, \phi, \sin \beta_0, n)$$

$$\frac{\partial D_h}{\partial \phi'} = 0.$$

An illustration of the wedge geometry is given in Figure 55.

FLOW DIAGRAM



SYMBOL DICTIONARY

BETN DIFFERENCE BETWEEN DIFFRACTION AND INCIDENCE ANGLE
 BETP DIFFERENCE BETWEEN DIFFRACTION AND IMAGE OF
 INCIDENCE ANGLE
 DH EDGE DIFFRACTION COEFFICIENT FOR THE HARD BOUNDARY
 CASE
 DIN INCIDENT PART OF EDGE DIFFRACTION COEFFICIENT
 DIP REFLECTION PART OF EDGE DIFFRACTION COEFFICIENT
 DPH SLOPE DIFFRACTION COEFFICIENT FOR THE HARD BOUNDARY CASE
 DPN INCIDENT PART OF SLOPE DIFFRACTION COEFFICIENT
 DPP REFLECTION PART OF SLOPE DIFFRACTION COEFFICIENT
 DPS SLOPE DIFFRACTION COEFFICIENT FOR THE SOFT BOUNDARY CASE
 DS EDGE DIFFRACTION COEFFICIENT FOR THE SOFT BOUNDARY
 CASE
 FN WEDGE ANGLE NUMBER
 LSURF A LOGICAL VARIABLE THAT IS SET TRUE IF THE SOURCE
 IS MOUNTED ON THE SURFACE OF THE WEDGE (GRAZING
 INCIDENCE)
 PH DIFFRACTED RAY PHI ANGLE IN DEGREES
 PHP INCIDENT RAY PHI ANGLE IN DEGREES
 R DISTANCE PARAMETER
 SBO SIN(BO)

CODE LISTING

```

1 C-----
2 SUBROUTINE DW(DS,DH,DPS,DPH,R,PH,PHP,SBO,FN,LSURF)
3 C!!!
4 C!!! WEDGE DIFFRACTION AND SLOPE DIFFRACTION COEFFICIENT
5 C!!! FOR THE SOFT AND HARD BOUNDARY CONDITIONS
6 C!!!
7 LOGICAL LSURF
8 COMPLEX DIN,DIP,DPN,DPP,DS,DH,DPS,DPH
9 C!!! INCIDENT PART OF DIFFRACTION COEFFICIENT
10 BETN=PH-PHP
11 CALL DI(DIN,R,BETN,SBO,FN,1...FALSE.)
12 CALL DPI(DPN,R,BETN,SBO,FN)
13 IF(.NOT.LSURF)GO TO 10
14 C!!! GRAZING INCIDENCE CASE
15 DS=(0.,0.)
16 DH=DIN
17 DPS=DPN
18 DPH=(0.,0.)
19 RETURN
20 10 CONTINUE
21 C!!! REFLECTION PART OF DIFFRACTION COEFFICIENT
22 BETP=PH+PHP
23 CALL DI(DIP,R,BETP,SBO,FN,1...FALSE.)
24 CALL DPI(DPP,R,BETP,SBO,FN)
25 DS=DIN-DIP
26 DH=DIN+DIP
27 DPS=DPN+DPP
28 DPH=DPN-DPP
29 RETURN
30 END
  
```

DZ

PURPOSE

To compute the diffraction coefficient for an edge formed by two curved surfaces.

METHOD

This subroutine computes the diffraction coefficient for a curved edge based on the uniform Geometrical Theory of Diffraction [4]. The diffraction coefficient is given by

$$D_{sh}(\phi, \phi', \beta_0) = \frac{e^{-j\pi/4}}{2n\sqrt{2\pi k} \sin\beta_0} \left[\frac{2 \sin(\pi/n) F[kL^1 a(\phi-\phi')]}{\cos(\pi/n) - \cos[(\phi-\phi')/n]} \right] \\ \pm \left\{ \cot\left(\frac{\pi+(\phi+\phi')}{2n}\right) F[kL^{rn} a^+(\phi+\phi')] \right. \\ \left. + \cot\left(\frac{\pi-(\phi+\phi')}{2n}\right) F[kL^{ro} a(\phi+\phi')] \right\},$$

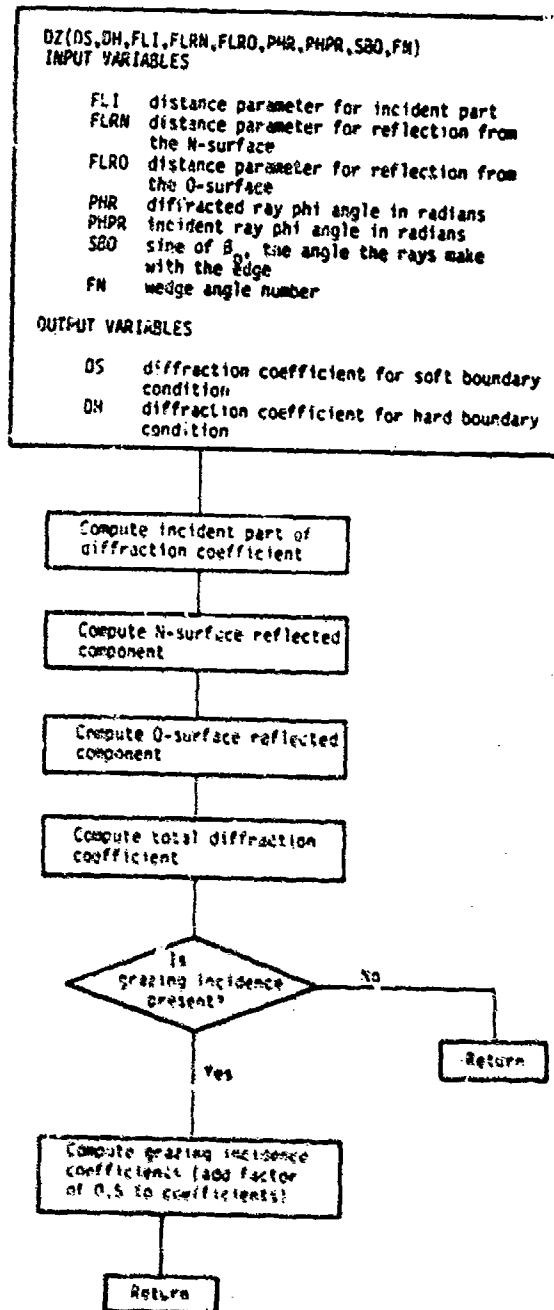
where $a(\beta) = 2 \cos^2 \beta/2$, $a^+(\beta) = 2 \cos^2(2\pi n - \beta)/2$, n is the wedge number (FN), and L, L^{rn}, L^{ro} are the distance parameters for the incident part, reflection from the n -surface and o -surface, respectively.

When the diffraction angle is close to one of the shadow boundaries, the following approximation is used

$$\cot\left(\frac{\pi-\beta}{2n}\right) F[kL a^+(\beta)] = \pm \sqrt{2\pi k L} e^{j\pi/4} e^{jk|L|a^+}.$$

where the plus or minus sign is chosen depending on which side of the shadow boundary the diffraction angle is on.

FLOW DIAGRAM



SYMBOL DICTIONARY

A	ANGLE FUNCTION FOR INCIDENT AND O-SURFACE TRANSITION FUNCTIONS
AP	ANGLE FUNCTION FOR N-SURFACE TRANSITION FUNCTION
CSP	$\cos(\text{PMR}/2.)$
DH	DIFFRACTION COEFFICIENT FOR HARD BOUNDARY CONDITION
DS	DIFFRACTION COEFFICIENT FOR SOFT BOUNDARY CONDITION
F1	CONSTANT FACTOR
F2	INCIDENT PART OF DIFFRACTION COEFFICIENT
F3	N-SURFACE PART OF DIFFRACTION COEFFICIENT
F4	O-SURFACE PART OF DIFFRACTION COEFFICIENT
FLI	DISTANCE PARAMETER FOR THE INCIDENT COMPONENT
FLRN	DISTANCE PARAMETER FOR THE REFLECTION FROM THE N-SURFACE
FLWO	DISTANCE PARAMETER FOR THE REFLECTION FROM THE O-SURFACE
FN	WEDGE ANGLE NUMBER
PHPH	INCIDENT RAY ANGLE IN RADIAN
PHN	DIFRACTED RAY ANGLE IN RADIAN
PMH	DIFFERENCE BETWEEN DIFFRACTION ANGLE AND THE INCIDENCE ANGLE
PPH	DIFFERENCE BETWEEN DIFFRACTION ANGLE AND THE IMAGE OF THE INCIDENCE ANGLE
SBO	SINE OF θ_0
TAN1	N-SURFACE ANGULAR DEPENDENCE OF DIFFRACTION COEFFICIENT
TAN2	O-SURFACE ANGULAR DEPENDENCE OF DIFFRACTION COEFFICIENT

CODE LISTING

```

1 C-----
2 SUBROUTINE DZ(DS,DH,FLI,FLRN,FLRO,PHR,PHPR,SBO,FN)
3 C!!!
4 C!!! CURVED EDGE DIFFRACTION COEFFICIENT
5 C!!!
6 COMPLEX FKY,F1,F2,F3,F4,DS,DH,CJ
7 COMMON/PI5/PI,TPI,DPR,RPD
8 PPR=PHR+PHPR
9 PMR=PIR-PHPR
10 F1=CEXP(CMPLX(0.,-PI/4.))/(2.*FN*TPI*SBO)
11 C!!! INCIDENT PART
12 CSP=COS(.5*PMR)
13 A=2.*CSP*CSP
14 IF(ABS(PMR-PI).LT.1.E-5) GO TO 10
15 F2=CMPLX(COS(PI/FN)-COS(PMR/FN),0.)
16 F2=2.*SIN(PI/FN)*FKY(FLI,A)/F2
17 GO TO 15
18 10 F2=CEXP(CMPLX(0.,PI/4.+TPI*ABS(FLI)*A))
19 IF(CSP.LT.0.) F2=-F2
20 F2=-F2*FN*TPI*CSORT(CMPLX(FLI,0.))
21 C!!! N-SURFACE REFLECTION PART
22 15 CSP=COS(.5*(TPI*FN-PPR))
23 AP=2.*CSP*CSP
24 TAN1=TAN((PI+PPR)/(2.*FN))
25 IF(ABS(TAN1).LT.1.E-5) GO TO 20
26 F3=FKY(FLRN,AP)/TAN1
27 GO TO 25
28 20 F3=CEXP(CMPLX(0.,PI/4.+TPI*ABS(FLRN)*AP))
29 IF(CSP.LT.0.) F3=-F3
30 F3=-F3*FN*TPI*CSORT(CMPLX(FLRN,0.))
31 C!!! C-SURFACE REFLECTION PART
32 25 CSP=COS(.5*PPR)
33 A=2.*CSP*CSP
34 TAN2=TAN((PI-PPR)/(2.*FN))
35 IF(ABS(TAN2).LT.1.E-5) GO TO 30
36 F4=FKY(FLRO,A)/TAN2
37 GO TO 35
38 30 F4=CEXP(CMPLX(0.,PI/4.+TPI*ABS(FLRO)*A))
39 IF(CSP.LT.0.) F4=-F4
40 F4=-F4*FN*TPI*CSORT(CMPLX(FLRO,0.))
41 C!!! TOTAL DIFFRACTION COEFFICIENT
42 35 DS=F1*(F2+F3+F4)
43 DH=F1*(F2-F3-F4)
44 C!!! GRAZING INCIDENCE CASE
45 IF(PPR.GT.1.E-5) GO TO 40
46 DS=.5*DS
47 DH=.5*DH
48 40 CONTINUE
49 RETURN
50 END

```

ENDIF

PURPOSE

To compute the fields due to the diffraction of source fields from a given cylinder end cap edge.

PERTINENT GEOMETRY

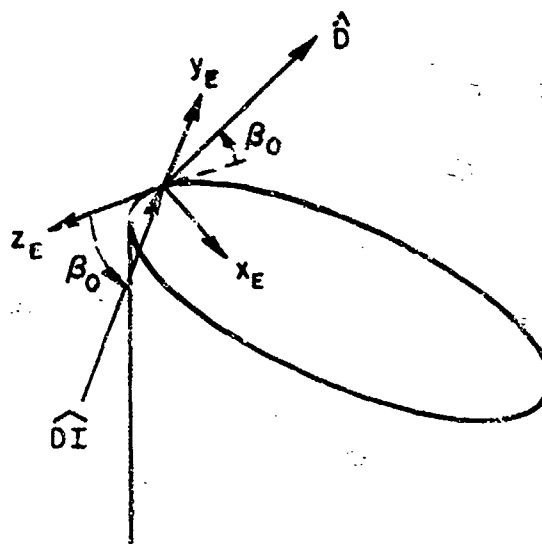


Figure 61-- Illustration of diffraction point coordinate system.

$$\hat{x}_E = \hat{x} XEX + \hat{y} XEY + \hat{z} XEZ$$

$$\hat{y}_E = \hat{x} YEX + \hat{y} YEY + \hat{z} YEZ$$

$$\hat{z}_E = \hat{x} ZEX + \hat{y} ZEY + \hat{z} ZEZ$$

METHOD

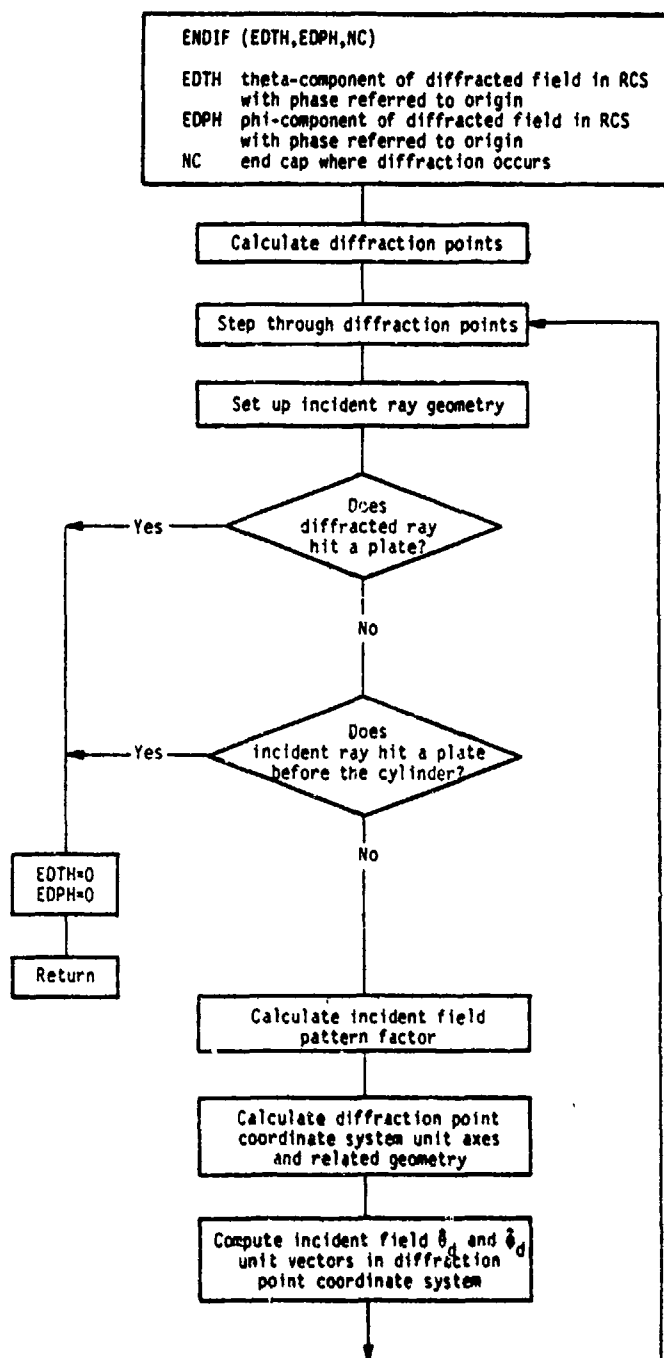
The Geometrical Theory of Diffraction [4] is used to compute the fields diffracted by the curved edges formed by the end cap disk and the curved surface of the elliptic cylinder. The form of diffraction coefficients for the curved edge are similar to that given in subroutine DIFPLT except that the distance parameters and spread factors are slightly different. The details are given on pages 127-131 of Reference 1. The fields from four possible diffraction points on the edge are superimposed to give the total diffracted field from one end cap. For small regions of the radiation pattern, it is possible that three of the diffraction points will coalesce into one point leaving two diffraction points on the edge. When

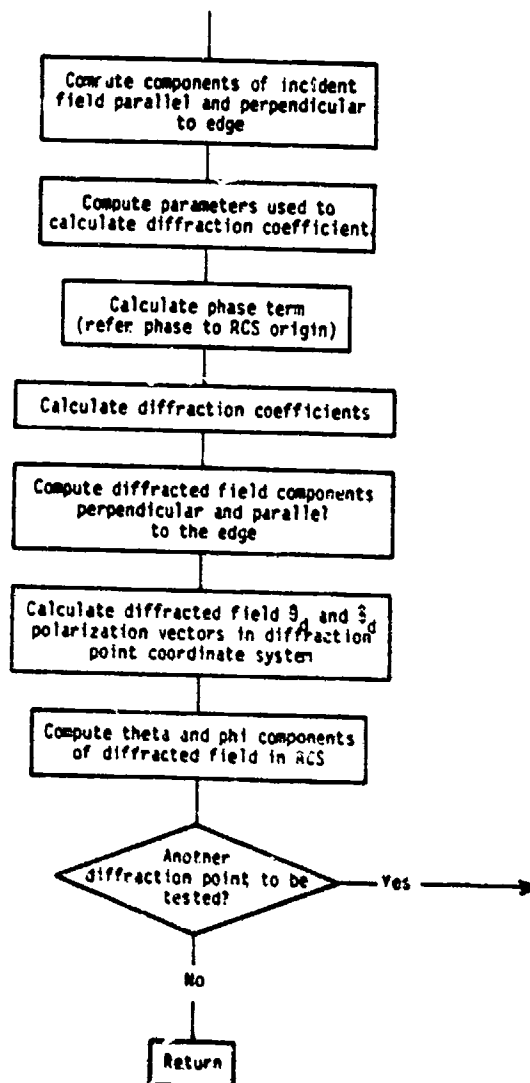
this happens a finite spike (psuedo caustic) of small angular extent appears in the pattern. One way to correct for this is by the use of an equivalent current solution[12]. However, this is costly in terms of computation time so it has not been included at present. The overall solution is not effected significantly by this approximation. The phases of the diffracted fields are referred to the reference coordinate system origin and the total field are represented as

$$E_{\text{endcap}}^d = w_m (EDTH\hat{\theta} + EDPH\hat{\phi}) \frac{e^{-jkR}}{R} ,$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight (w_m) are added elsewhere in the code.

FLOW DIAGRAM





SYMBOL DICTIONARY

AE	RADIUS OF CURVATURE OF EDGE AT DIFFRACTION POINT IN END CAP PLANE
BO	THE ANGLE THE INCIDENT (AND DIFFRACTED) RAY MAKES WITH THE EDGE UNIT VECTOR
CBO	COSINE OF BO (DOT PRODUCT OF DIFF RAY AND Z AXIS OF DIFFRACTION POINT COORD SYS)
CPE	COSINE OF PHER
CTE	COSINE OF THER
CTHI	DOT PRODUCT OF INCIDENT RAY PROPAGATION DIRECTION UNIT VECTOR AND CYLINDER UNIT NORMAL
CV	COSINE OF VR
D	X,Y,Z COMPONENTS OF PROPAGATION DIRECTION AFTER DIFFRACTION IN RCS
DH	DIFFRACTION COEF FOR HARD BOUNDARY CONDITION
DHIT	DISTANCE FROM SOURCE TO NEAREST HIT (FROM PLAIN)
DI	X,Y,Z COMPONENTS OF UNIT VECTOR OF INCIDENT RAY PROPAGATION DIRECTION IN RCS
DS	DIFFRACTION COEF. FOR SOFT BOUNDARY CONDITION
EDPH	PHI COMPONENT OF DIFFRACTED E FIELD IN RCS WITH PHASE REFERRED TO RCS ORIGIN
EDPP	COMPONENT OF DIFFRACTED FIELD PARALLEL TO EDGE
EDPR	COMPONENT OF DIFFRACTED FIELD PERPENDICULAR TO EDGE
EDTH	THETA COMPONENT OF DIFFRACTED E FIELD IN RCS WITH PHASE REFERRED TO RCS ORIGIN
EF	THETA COMPONENT OF INCIDENT FIELD PATTERN FACTOR IN RCS
EG	PHI COMPONENT OF INCIDENT FIELD PATTERN FACTOR IN RCS
EIPP	COMPONENT OF INCIDENT E FIELD PARALLEL TO EDGE
EIPR	COMPONENT OF INCIDENT E FIELD PERPENDICULAR TO EDGE
EIX	X,Y,Z COMPONENTS OF INCIDENT FIELD PATTERN FACTOR
EIY	
EIZ	
EM	NORMALIZATION CONSTANT FOR Z AXIS OF DIF POINT COORD SYS
EX	X,Y,Z COMPONENTS DEFINING UNIT EDGE VECTOR (Z AXIS OF DIFFRACTION POINT COORD SYS)
EY	
EZ	
FN	WEDGE ANGLE NUMBER
I	DO LOOP VARIABLE
LHIT	SET TRUE IF RAY HITS A PLATE (FROM PLAIN)
NC	END CAP WHERE DIFFRACTION OCCURS
NCC	SIGN CHANGE VARIABLE
PH	COMPLEX PHASE COEFFICIENT
PHEDN	PHI COMPONENT OF DIFFRACTED RAY DIRECTION IN DIFFRACTION POINT COORDINATE SYSTEM
PHER	PHI COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN DIFFRACTION POINT COORDINATE SYSTEM
PHEX	POLARIZATION UNIT VECTOR IN PHI DIRECTION FOR INC. OR DIFFRACTED RAY IN DIFFRACTION POINT COORDINATE SYSTEM IN (X,Y,Z) RCS COMPONENTS
PHEY	
PHEZ	
PHIR	PHI COMPONENT OF INCIDENT RAY DIRECTION IN RCS
RG	RADIUS OF CURVATURE OF CYLINDER SURFACE AT DIFF POINT IN X-Y PLANE
KGAE	RADIUS OF CURVATURE OF EDGE AT DIFFRACTION POINT IN END CAP PLANE
SBO	SINE OF BO
SPE	SINE OF PHER
SPA	X,Y,Z COMPONENTS OF UNIT VECTOR OF PROPAGATION DIRECTION OF INCIDENT RAY
SPY	
SPZ	
SSBO	SINE OF BO SQUARED
STE	SINE OF THER

SV SINE OF VR
 T1 } X,Y,Z COMPONENTS DEFINING THE INCIDENT (OR DIFF)
 T2 } RAY PROPAGATION DIRECTION IN DIFFRACTION
 T3 } POINT COORD SYSTEM
 THENH THETA COMPONENT OF DIFFRACTED RAY DIRECTION IN
 DIFFRACTION POINT COORDINATE SYSTEM
 THER THETA COMPONENT OF INCIDENT RAY PROPAGATION
 DIRECTION IN DIFFRACTION POINT COORDINATE SYSTEM
 THEX } POLARIZATION UNIT VECTOR IN THETA DIRECTION
 THEY } FOR INCIDENT OR DIFFRACTED RAY IN DIFFRACTION
 THEZ } POINT COORD SYSTEM IN (X,Y,Z) RCS COMPONENTS
 THIR } THETA COMPONENT OF INCIDENT RAY DIRECTION IN RCS
 TOP COMPUTATIONAL VARIABLE
 UB X,Y,Z COMPONENTS OF UNIT VECTOR TANGENT TO
 CYLINDER AT DIFFRACTION POINT (2-D)
 UN X,Y,Z COMPONENTS OF UNIT NORMAL TO CYLINDER
 AT DIFFRACTION POINT (2-D)
 UNEM NORMALIZATION CONSTANT FOR EDGE UNIT NORMAL NE
 UNEX }
 UNEY } X,Y,Z COMPONENTS OF UNIT NORMAL TO EDGE IN
 UNEZ } END CAP PLANE IN RCS
 V ELL ANGLES DEFINING (UP TO) 4 DIFFRACTION POINTS
 ON END CAP NC
 VR ELL ANGLE DEFINING DIFFRACTION POINT IN ERCS
 VXS X,Y,Z COMPONENTS OF UNIT VECTORS DEFINING SOURCE
 COORDINATE SYSTEM AXES DIRECTIONS IN RCS
 XC X,Y,Z COMPONENTS OF DIFFRACTION POINT LOCATION
 IN RCS
 XEX } X,Y,Z COMPONENTS DEFINING UNIT VECTOR OF X
 XEY } AXIS OF DIFFRACTION POINT COORDINATE SYSTEM
 XEZ } (VECTOR NORMAL TO EDGE AND PARALLEL TO END CAP
 PLANE)
 YEX } X AND Z COMPONENTS DEFINING UNIT VECTOR OF Y AXIS
 YEZ } OF DIFF. POINT COORD SYS (VECTOR NORMAL TO END CAP)

CODE LISTING

```

1 C-----
2 SUBROUTINE ENDIF(EDTH,EDPH,NC)
3 C!!!
4 C!!! COMPUTES THE DIFFRACTED FIELD FROM THE END CAP RIM
5 C!!!
6 COMPLEX EDTH,EDPH,EIX,EIY,EIZ,EIPR,EIPP,PH,EDPR,EDPP,DS,DH
7 COMPLEX CJ,CPI4,EF,EG
8 DIMENSION V(4),UN(2),UB(2),DI(3),XC(3)
9 LOGICAL LHIT,LDEBUG,LTEST
10 COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
11 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
12 COMMON/SORINF/XS(3),VXS(3,3)
13 COMMON/COMP/CJ,CPI4
14 COMMON/PIS/PI,TPI,DPR,RPD
15 COMMON/THPHUV/DT(3),DP(2)
16 COMMON/TEST/LDEBUG,LTEST
17 EDTH=(0.,0.)
18 EDPH=(0.,0.)
19 IF(LDEBUG) WRITE(6,900)
20 C!!! CALCULATE DIFFRACTION POINTS
21 900 FORMAT(/,' DEBUGGING ENDIF SUBROUTINE')
22 CALL DFPTCL(V,NC)
23 IF(LDEBUG) WRITE(6,*) NC,V
24 C!!! STEP THRU DIFFRACTION POINTS
25 DO 1 I=1,4
26 IF(V(I).LT.-500.) GO TO 2
27 C!!! SET UP INCIDENT RAY GEOMETRY
28 VR=V(I)*RPD
29 SV=SIN(VR)
30 CV=COS(VR)
31 XC(1)=A*CV
32 XC(2)=B*SV
33 XC(3)=A*(CTC(NC)*CV+ZC(NC))
34 C!!! DOES DIFFRACTED RAY HIT A PLATE?
35 CALL PLAIN(XC,D,DHIT,0,LHIT)
36 IF(LHIT) GO TO 1
37 SPX=XC(1)-XS(1)
38 SPY=XC(2)-XS(2)
39 SPZ=XC(3)-XS(3)
40 SPM=SQRT(SPX*SPX+SPY*SPY+SPZ*SPZ)
41 SPX=SPX/SPM
42 SPY=SPY/SPM
43 SPZ=SPZ/SPM
44 TOP=SQRT(SPX*SPX+SPY*SPY)
45 THIR=BTAN2(TCP,SPZ)
46 PHIR=BTAN2(SPY,SPX)
47 DI(1)=SPX
48 DI(2)=SPY
49 DI(3)=SPZ
50 C!!! DOES INCIDENT RAY HIT PLATE BEFORE END CAP?
51 CALL PLAIN(XS,DI,DHIT,0,LHIT)
52 IF(LHIT.AND.(DHIT.LT.SPM)) GO TO 1
53 C!!! CALCULATE INCIDENT FIELD PATTERN FACTOR
54 CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
55 IF(LDEBUG) WRITE(6,*) EF,EG
56 EX=-A*SV
57 EY=B*CV
58 EZ=-A*CTC(NC)*SV
59 EN=SQRT(EX*EX+EY*EY+EZ*EZ)
60 NCC=NC
61 IF(NCC.GT.1)NCC=-1
62 C!!! CALCULATE DIF. POINT COORD. SYS UNIT AXES AND RELATED GEOM.
63 EX=NCC*EX/EN
64 EY=NCC*EY/EN
65 EZ=NCC*EZ/EN
66 CBO=D(1)*EX+D(2)*EY+D(3)*EZ

```

```

07      IF(CBO.GT.1.) CBO=1.
08      SBO=SQRT(1.-CBO*CBO)
09      SSBO=SBO*SBO
10      UNEX=B*CV*SNC(NC)
11      UNEY=A*SV/SNC(NC)
12      UNEZ=B*CNC(NC)*CV
13      UNEM=SQRT(UNEX*UNEX+UNEY*UNEY+UNEZ*UNEZ)
14      UNEX=UNEX/UNEM
15      UNEY=UNEY/UNEM
16      UNEZ=UNEZ/UNEM
17      RG=((A*A*SV*SV+B*B*CV*CV)**(1.5))/A/B
18      RGAE=A*A*SV*SV+B*B*SNC(NC)*SNC(NC)*CV*CV
19      HGAE=(RG/AE**1.5)/A/B
20      AE=HGAE/SNC(NC)/SNC(NC)
21      CALL NANDB(UN,UB,VR)
22      YEX=-CNC(NC)*NCC
23      YEZ=SNC(NC)*NCC
24      XEX=-YEZ*EY
25      XEY=YEZ*EX-YEX*EZ
26      XEZ=YEX*EY
27      T1=XEX*SPX+XEY*SPY+XEZ*SPZ
28      T2=YEX*SPX+YEZ*SPZ
29      T3=EX*SPX+EY*SPY+EZ*SPZ
30      THER=BTAN2(SQRT(T1*T1+T2*T2),-T3)
31      PHER=BTAN2(-T2,-T1)
32      IF(PHER.LT.0.) PHER=TP1+PHER
33      FN=1.+ACOS(UN(1)*YEX)/PI
34      IF(PHER.GT.FN*PI)GO TO 1
35      CTE=COS(THER)
36      STE=SIN(THER)
37      CPE=COS(PHER)
38      SPE=SIN(PHER)
39      CIII CALCULATE INCIDENT FIELD THETA AND PHI POLARIZATION
100 CIII UNIT VECTORS
101      ITEX=XEX*CTE*CPE+YEX*CTE*SPE-EX*STE
102      THEY=XEY*CTE*CPE+EY*STE
103      ITEZ=XEZ*CTE*CPE+YEZ*CTE*SPE-EZ*STE
104      PHEX=-XEX*SPE+YEX*CPE
105      PHEY=-XEY*SPE
106      PHEZ=-XEZ*SPE+YEZ*CPE
107 CIII COMPUTE COMPONENTS OF INC. FIELD PERPENDICULAR AND PARALLEL
108 CIII TO THE EDGE
109      EIPR=EIX*PHEX+EIY*PHEY+EIZ*PHEZ
110      EIPP=EIX*THEX+EIY*THEY+EIZ*THEZ
111 CIII COMPUTE PARAMETERS USED IN DIF. COEF. CALCULATIONS
112      T1=UNEX*(SPX-D(1))+INEY*(SPY-D(2))+UNEZ*(SPZ-D(3))
113      R=SPM*AE*SSBO/(AE*SSBO-T1*SPM)
114      FLI=SPM*SSBO
115      FLRO=SPM*SSBO
116      TI=UN(1)*UNEX+UN(2)*UNEY
117      CTHI=-(SPX*UN(1)+SPY*UN(2))
118      HRN=SPM*AE*SSBO/(AE*SSBO+2*T1*CTHI*SPM)
119      WR=BTAN2(-SPX*UB(1)-SPY*UB(2),-SPZ)
120      SSN=SIN(WR)**2
121      SCN=COS(WR)**2
122      SST2=SSN*SCN*CTHI*QTHI
123      RHO2=SPM
124      RHO1=SPM*HG*CTHI/(HO*CTHI+2.*SPM*SST2)
125      IF(CTHI.LT.1.E-5)RHO1=SPM
126      FLRN=RHO1*RHO2*SSBO/HRN
127      T1=XEX*D(1)+XEY*D(2)+XEZ*D(3)
128      T2=YEX*D(1)+YEZ*D(3)
129      T3=EX*D(1)+EY*D(2)+EZ*D(3)
130      THEDR=BTAN2(SQRT(T1*T1+T2*T2),T3)
131      PHEDR=BTAN2(T2,T1)
132      IF(PHEDR.LT.0.) PHEDR=TP1+PHEDR

```

```

133      IF(PHEDR.GT.FN*PI)GO TO 1
134      CTE=COS(THEDR)
135      STE=SIN(THEDR)
136      CPE=COS(PHEDR)
137      SPE=SIN(PHEDR)
138 C!!!  CALCULATE PHASE TERM
139      PH=CEXP(-CJ*TPI*SPH)/SPH
140      PH=PH*CEXP(CJ*TPI*(XC(1)*D(1)+XC(2)*D(2)+XC(3)*D(3)))
141 C!!!  CALCULATE DIFFRACTION COEFFICIENTS
142      CALL DZ(DS,DH,FLI,FLRN,FLRO,PHEDR,PHER,SBO,FN)
143      IF(LDEBUG) WRITE(6,*) FLI,FLRN,FLRO,PHEDR,PHER,SBO,FN
144      IF(LDEBUG) WRITE(6,*) DS,DH
145      IF(R.GE.0.) GO TO 5
146      R=ABS(R)
147      PH=(0.,1.)*PH
148 5      CONTINUE
149 C!!!  CALCULATE DIF. FIELD COMPONENTS PERPENDICULAR AND PARALLEL
150 C!!!  TO THE EDGE
151      EDPR=-DH*SQRT(R)*EIPR*PH
152      EDPP=-DS*SQRT(R)*EIPP*PH
153 C!!!  CALCULATE DIF. FIELD THETA AND PHI POLARIZATION UNIT VECTORS
154      THEX=XEX*CTE*CPE+YEX*CTE*SPE-EX*STE
155      THEY=XEY*CTE*CPE-EY*STE
156      THEZ=XEZ*CTE*CPE+YEZ*CTE*SPE-EZ*STE
157      THEX=-THEX
158      THEY=-THEY
159      THEZ=-THEZ
160      PHEX=-XEX*SPE+YEX*CPE
161      PHEY=-XEY*SPE
162      PHEZ=-XEZ*SPE+YEZ*CPE
163 C!!!  CALCULATE THETA AND PHI COMPONENTS OF DIF. FIELD IN RCS
164      EDTH=EDTH+EDPR*(PHEX*DT(1)+PHEY*DT(2)+PHEZ*DT(3))
165      EDTH=EDTH+EDPP*(THEX*DT(1)+THEY*DT(2)+THEZ*DT(3))
166      EDPH=EDPH+EDPR*(PHEX*DP(1)+PHEY*DP(2))
167      EDPH=EDPH+EDPP*(THEX*DP(1)+THEY*DP(2))
168 1      CONTINUE
169 2      IF(.NOT.LTEST) RETURN
170      WRITE(6,410)
171 410    FORMAT(/,' TESTING ENDIF SUBROUTINE')
172      WRITE(6,*) EDTH,EDPH,NC
173      RETURN
174      END

```

FCT

PURPOSE

This function computes the integrand for various integrals used to compute the diffraction coefficient for an elliptic cylinder.

METHOD

For the present code, only the integrand defined for ID equal to three is used. This is used to define the arc length between two points on the elliptic cylinder. The arc length is given by

$$t = \frac{1}{|\sin \alpha_s|} \int_{v_i}^{v_f} FCT(v) dv ,$$

where

$$FCT(x) = \sqrt{A^2 \sin^2 x + B^2 \cos^2 x} .$$

SYMBOL DICTIONARY

A2	THE SQUARE OF THE RADIUS OF THE ELLIPTIC CYLINDER ON THE X-AXIS
B2	THE SQUARE OF THE RADIUS OF THE ELLIPTIC CYLINDER ON THE Y-AXIS
CN	COSINE OF X
F	SQRT((A*SIN(VR))**2+(B*COS(VR))**2)
SN	SINE OF X
SNA	THE ABSOLUTE VALUE OF THE SINE OF THE ANGLE MEASURED FROM THE NEGATIVE Z-AXIS OF THE CYLINDER TO THE DIRECTION OF PROPAGATION
X	THE ARGUMENT OF THE INTEGRAND DEFINING THE ELLIPTIC ANGLE

CODE LISTING

```

1 C-----
2      FUNCTION FCT(X)
3 C!!!
4 C!!!  THESE ARE INTEGRAND OF ATTENUATION COEFFICIENT INTEGRATION.
5 C!!!
6      COMMON/GEOMEL/A,B,ZC(2),SIC(2),CNC(2),CTC(2)
7      COMMON/PI S/PI,TP1,DPR,RPD
8      COMMON/GID/A5,ID,SAS,SASP,CAS
9      A2 = A*A
10     B2 = B*B
11     SNA=ABS(SAS)
12     SN = SIN(X)
13     CS = COS(X)
14     SN2 = SIN (2.*X)
15     CS2 = COS(2.*X)
16     Q = (A2*B2*SNA)**(1./3.)
17     GIN = 3.*(A2-B2)/Q
18     F = SQRT (A2*SN*SN+B2*CS*CS)
19     IF (ID .EQ. 3) GO TO 3
20     IF (ID .EQ. 2) GO TO 2
21     IF (ID .EQ. 4) GO TO 4
22     IF (ID .EQ. 5) GO TO 5
23     FCT = Q/F
24     RETURN
25 2    FCT = Q*CS*SNA/(F*F*F)
26     RETURN
27 3    FCT = F
28     RETURN
29 4    FCT = GIN*CS2/F
30     RETURN
31 5    FCT = .75*(A2-B2)*GIN*SN2*SN2/F/F/F
32     RETURN
33     END

```

FFCT

PURPOSE

The purpose of this function is to determine the transition function for the edge and corner diffraction coefficients.

METHOD

The transition function for the edge and corner diffraction coefficients is given by[4]:

$$\text{FFCT}(x) = 2j|\sqrt{x}| e^{jx} \int_{|\sqrt{x}|}^{\infty} e^{-j\tau^2} d\tau.$$

This can also be written as

$$\text{FFCT}(x) = j\sqrt{2\pi|x|} e^{jx} \left[(0.5 - j0.5) - \left(C\left(\sqrt{\frac{2|x|}{\pi}}\right) - jS\left(\sqrt{\frac{2|x|}{\pi}}\right) \right) \right]$$

where

$$\int_0^{\alpha} e^{-j\frac{\pi}{2}t^2} dt = C(\alpha) - jS(\alpha).$$

SYMBOL DICTIONARY

CFR	REAL PART OF FRESNEL INTEGRAL
DEL	ARGUMENT OF TRANSITION FUNCTION
FFCT	TRANSITION FUNCTION
S	ARGUMENT OF FRESNEL INTEGRAL
SDEL	SORT(ABS(DEL))
SPR	IMAGINARY PART OF FRESNEL INTEGRAL

CODE LISTING

```
1 C-----
2      COMPLEX FUNCTION FFCT(DEL)
3 CIII
4 CIII  DETERMINES THE TRANSITION FUNCTION RESULT FOR THE EDGE
5 CIII  AND CORNER DIFFRACTION COEFFICIENTS.
6 CIII
7      COMMON/PI5/PI,TP1,OPR,RPD
8      IF(ABS(DEL).GT.10.) GO TO 1
9      SDEL=SQRT(ABS(DEL))
10     S=SQRT(2./PI)*SDEL
11     CALL FNNEL(CFR,SPR,S)
12     FFCT=CMPLX(0.5-CFR,SPR-0.5)
13     FFCT=SQRT(TP1)*SDEL*FFCT*CEXP(CMPLX(0.,DEL*PI/2.))
14     RETURN
15 1    FFCT=(1.,0.)
16     RETURN
17     END
```

FKARG

PURPOSE

To compute a parameter needed in the diffraction coefficient for the elliptic cylinder.

METHOD

This subroutine computes the parameter used in the diffraction coefficient to determine the fields scattered from the elliptic cylinder. This parameter is given by [6],

$$\xi = \int_{Q_1}^{Q_2} \pi^{1/3} \rho_g^{-2/3} dt,$$

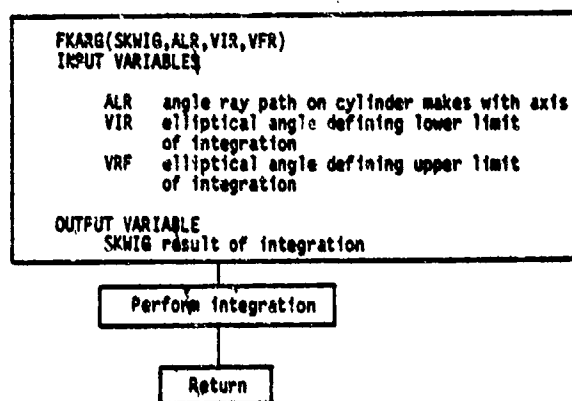
where ρ_g is the radius of curvature of the elliptic cylinder in the plane of propagation. This can also be written as

$$\xi = \pi^{1/2} (AB)^{2/3} |\sin \alpha|^{1/3} \int_{v_i}^{v_f} \frac{dv}{\sqrt{A^2 \sin^2 v + B^2 \cos^2 v}},$$

where

ξ = SKWIG
 α = ALR
 v_i = VIR
 v_f = VFR.

FLOW DIAGRAM



SYMBOL DICTIONARY

ALR ANGLE MEASURED FROM NEGATIVE Z-AXIS IN THE DIRECTION
 OF PROPAGATION
 ANS THE EVALUATED INTEGRAL
 FUNI INTEGRAND OF THE INTEGRAL
 SKWIG PARAMETER USED TO DEFINE CURVED SURFACE AT THE POINT
 OF DIFFRACTION
 VFR ELLIPTICAL ANGLE DEFINING THE DIFFRACTION ANGLE
 POSITION ON THE CYLINDER
 VIR ELLIPTICAL ANGLE DEFINING THE INCIDENT ANGLE POSITION
 ON CYLINDER

CODE LISTING

```

1 C-----
2        SUBROUTINE FKARG(SKWIG,ALR,VIR,VFR)
3 C!!!
4 C!!!    COMPUTES THE PARAMETER NEEDED IN THE DIFFRACTION
5 C!!!    COEFFICIENT FOR THE ELLIPTIC CYLINDER
6 C!!!
7        COMMON/PIS/PI,TPI,DPR,RPD
8        COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
9        EXTERNAL FUNI
10        IF (ABS(VIR-VFR).LT.1.E-5) GO TO 1
11        SKWIG=(PI*ABS(SIN(ALR)))**(1./3.)
12        SKWIG=SKWIG*((A*B)**(2./3.))
13        CALL DQCS2(VIR,VFR,FUNI,ANS)
14        SKWIG=SKWIG*ANS
15        SKWIG=ATN(SKWIG)
16        RETURN
17 1       SKWIG=0.
18        RETURN
19        END
  
```

FKY

PURPOSE

This function is used in computing the transition function for curved edge diffraction.

METHOD

The transition function for the diffraction coefficient of an edge in a curved surface is the same as for a straight wedge, except that the curved edge function takes into account the possibility of the distance parameter being negative. The transition function is given by [4]

$$F(x) = 2j\sqrt{x} e^{jx} \int_{\sqrt{x}}^{\infty} e^{-j\tau^2} d\tau,$$

where $x = kLa$,

and $k = 2\pi/\lambda$

L = distance parameter

a = a function dependent on the square of the cosine of the incident and diffraction angles and the wedge angle number.

The transition function can then be written as,

$$F(kLa) = j2\pi \sqrt{\frac{|L|a}{\lambda}} e^{jk|L|a} \left[(0.5 - j0.5) - \left(C\left(2\sqrt{\frac{|L|a}{\lambda}}\right) - jS\left(2\sqrt{\frac{|L|a}{\lambda}}\right) \right) \right],$$

for $L > 0$

and

$$F(kLa) = F^*(k|L|a), \quad \text{for } L < 0$$

where the "*" means the complex conjugate and

$$\int_0^{\alpha} e^{-j\frac{\pi}{2}t^2} dt = C(\alpha) - jS(\alpha).$$

The above equation relates to the function FKY as,

$$FKY(L/\lambda, a) = F(kLa).$$

SYMBOL DICTIONARY

A	PARAMETER DEPENDANT ON THE INCIDENT AND DIFFRACTED ANGLES
C	REAL PART OF FRESNEL INTEGRAL
FKY	TRANSITION FUNCTION
FL	THE DISTANCE PARAMETER IN WAVELENGTHS
FLA	ABSOLUTE VALUE OF FL
S	IMAGINARY PART OF FRESNEL INTEGRAL
XS	ARGUMENT OF FRESNEL INTEGRAL

CODE LISTING

```

1 C-----
2 FUNCTION FKY(FL,A)
3 C!!!
4 C!!! TRANSITION FUNCTION FOR CURVED EDGE DIFFRACTION
5 C!!!
6 COMPLEX FKY
7 COMMON/PIS/PI,TPI,DPR,RPD
8 FLA=ABS(FL)
9 XS=2.*SORT(FLA*A)
10 FKY=CMPLX(0.,TPI)*SORT(FLA*A)
11 FKY=FKY*CEXP(CMPLX(0.,TPI*FLA*A))
12 CALL FRNELS(C,S,XS)
13 FKY=FKY*CMPLX(.5-C,S-.5)
14 IF(FL.GE.0.) RETURN
15 FKY=CONJG(FKY)
16 RETURN
17 END

```

FRNELS

PURPOSE

To compute the Fresnel integral,

$$f(x_s) = \int_0^{x_s} e^{-j\pi/2 u^2} du = C(x_s) - j S(x_s) .$$

METHOD

The integral is evaluated using an approximation by J. Boersma [13].
The integral

$$f(x) = \int_0^x \frac{e^{-jt}}{\sqrt{2\pi t}} dt$$

is approximated as follows:

$$\text{for } 0 \leq x \leq 4 \quad f(x) \approx e^{-jx} \sqrt{\frac{x}{4}} \sum_{n=0}^{11} (a_n + jb_n) \left(\frac{x}{4}\right)^n$$

$$\text{for } x \geq 4 \quad f(x) \approx \frac{1-j}{2} + e^{-jx} \sqrt{\frac{4}{x}} \sum_{n=0}^{11} (c_n + jd_n) \left(\frac{4}{x}\right)^n$$

(the constants a_n , b_n , c_n and d_n are provided by Boersma and are defined in data statements in the subroutine).

Note that by performing a change of variable, the integral to be solved becomes of the form of the integral which Boersma solved;

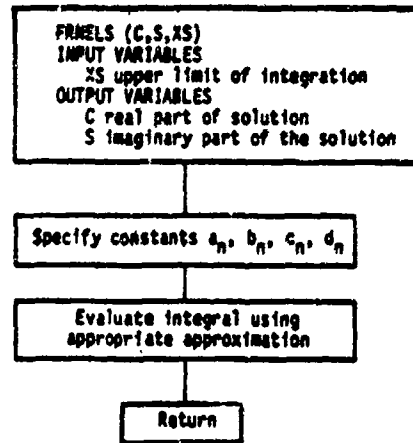
$$t = \frac{\pi}{2} u^2 .$$

By applying this change of variable, we get

$$f(x_s) = \int_0^{x_s} e^{-j\frac{\pi}{2} u^2} du = \int_0^x \frac{e^{-jt}}{\sqrt{2\pi t}} dt$$

$$\text{where} \quad x = \frac{\pi}{2} x_s^2 .$$

FLOW DIAGRAM



SYMBOL DICTIONARY

A	}	CONSTANTS USED IN EVALUATING INTEGRAL
B		
C		
D		
F _I	}	IMAGINARY COMPONENT OF SUMMATION FUNCTION
F _R		

CODE LISTING

```

1 C-----
2      SUBROUTINE FRNELS(C,S,XS)
3 CIII
4 CIII  THIS IS THE FRESNEL INTEGRAL SUBROUTINE WHERE THE INTEGRAL
5 CIII  IS FROM U=0 TO XS, THE INTEGRAND IS  $\exp(-j\pi/2 \cdot u \cdot u)$ ,
6 CIII  AND THE OUTPUT IS C(XS)-J*S(XS).
7 CIII
8      LOGICAL LDEBUG,LTEST
9      COMMON/TEST/LDEBUG,LTEST
10     COMMON/PI/PI,TPI,DPR,RPD
11     DIMENSION A(12),B(12),CC(12),D(12)
12 CIII  SPECIFY CONSTANTS
13     DATA A/1.595769140,-0.000001702,-0.000560054,-0.000576361,0.92069
14         2902,-0.016898657,-3.050485660,-0.075752419,0.850663781,-0.02
15         563904
16         21,-0.150230960,0.034404779/
17     DATA B/-0.000000033,4.255387524,-0.000092810,-7.780020400,-0.0095
18         20895,5.075161290,-0.130341947,-1.363729124,-0.403349276,0.70
19         222201
20         26,-0.216195929,0.019547031/
21     DATA CC/0,-0.024933975,0.000003936,0.005770956,0.000689892,-0.00
22         2497136,0.011948809,-0.006748873,0.000246420,0.002102967,-0.0
23         012174
24         230,0.000233939/
25     DATA D/0.199471140,0.000000023,-0.009351341,0.000023006,0.0048514
26         26,0.001403218,-0.017122914,0.029004067,-0.027928955,0.016497
27         308,-0.
28     2.005598515,0.000838386/
29     IF(XS.LE.0.0) GO TO 414
30     X=XS
31     X = PI*λ*X/2.0
32     FH=0.0
33     FI=0.0
34     K=13
35 CIII  IS X<4?
36     IF(X-4.0) 10,40,40
37     10  Y=X/4.0
38 CIII  EVALUATE INTEGRAL USING X<4 APPROXIMATION
39     K=K-1
40     FR=(FR+A(K))*Y
41     FI=(FI+B(K))*Y
42     IF(K-2) 30,30,20
43     FR=FR+A(1)
44     FI=FI+B(1)
45     C=(FR+COS(X)+FI*SIN(X))*SORT(Y)
46     S=(FR*SIN(X)-FI*COS(X))*SORT(Y)
47     GO TO 1
48 CIII  EVALUATE INTEGRAL USING X>4 APPROXIMATION
49     Y=4.0/X
50     K=K-1
51     FR=(FR+CC(K))*Y
52     FI=(FI+D(K))*Y
53     IF(K-2) 60,60,50
54     FR=FR+CC(1)
55     FI=FI+D(1)
56     C=0.5*(FR+COS(X)+FI*SIN(X))*SORT(Y)
57     S=0.5*(FR*SIN(X)-FI*COS(X))*SORT(Y)
58     GO TO 1
59 414  C=0.0
60     S=0.0
61     IF (.NOT.LTEST) GO TO 2

```

```
58      WRITE (6,3)
59  3    FORMAT (/,' TESTING FRNELS SUBROUTINE')
60      WRITE (6,*) C,S,XS
61  2    RETURN
62      END
```

FUNI

PURPOSE

This function calculates the integrand of the integral in subroutine FKARG.

METHOD

The integrand of this integral evaluated in subroutine FKARG is given by

$$\text{FUNI}(\text{VR}) = \frac{1}{\sqrt{A^2 \sin^2(\text{VR}) + B^2 \cos^2(\text{VR})}}$$

SYMBOL DICTIONARY

A	RADIUS OF CYLINDER ON X-AXIS
B	RADIUS OF CYLINDER ON Y-AXIS
VR	ELLIPTIC ANGLE ON CYLINDER IN RADIANS

CODE LISTING

```
1 C-----
2      FUNCTION FUNI(VR)
3      C!!!
4      C!!! INTEGRAND OF INTEGRAL NEEDED IN FKARG
5      C!!!
6      COMMON/GEOMEL/A,B,ZC(2),SIC(2),CNC(2),CTC(2)
7      FUNI=1./SQRT(A*A*SIN(VR)*SIN(VR)+B*B*COS(VR)*COS(VR))
8      RETURN
9      END
```

GEOM

PURPOSE

This subroutine calculates a large number of constants that are fixed for a given geometry of plates. They are stored in common blocks for use in other sections of the program. It is called once for every source used. Because of the diversity of operations done in GEOM, its description is broken into seven parts:

1. Identify edges which are common to more than one plate.
2. Compute unit vectors of edge-fixed coordinate systems for each edge on each plate.
3. Determine source image information for reflections from plates.
4. Calculate possible range for diffraction angle β_0 for each edge.
5. Determine wedge angles for plates with common edges.
6. Determine plates which are totally shadowed from the source.
7. Perform calculations for plates which intersect.

GEOM, SECTION 1

PURPOSE

To identify edges which are common to two plates.

PERTINENT GEOMETRY

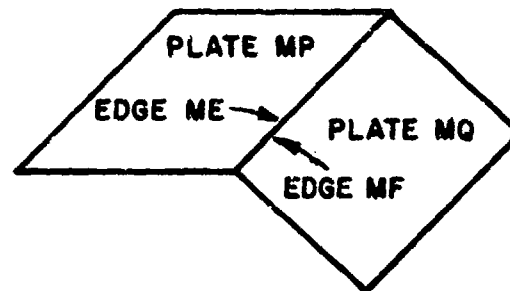
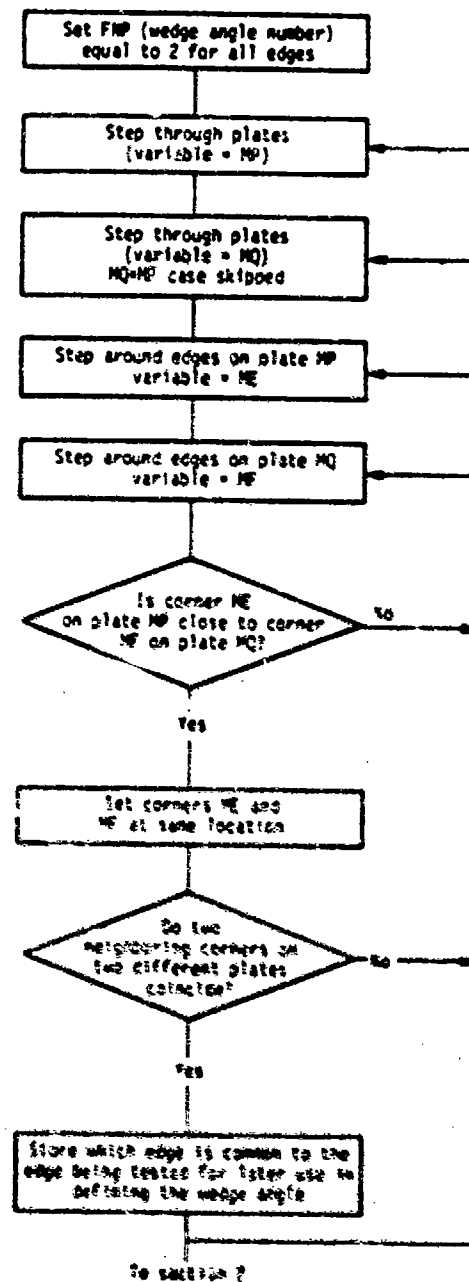


Figure 62--Illustration of two plates with a common edge.

FLOW DIAGRAM



CODE LISTING

```

1 C
2 SUBROUTINE GEOM
3 C!!!
4 C!!! THIS ROUTINE COMPUTES ALL THE GEOMETRY ASSOCIATED
5 C!!! WITH FIXED PLATE STRUCTURE, SUCH AS EDGE UNIT VECTORS,
6 C!!! PLATE NORMALS, SHADOWED PLATES, ETC.
7 C!!!
8 DIMENSION IHIT(6), XI(3), XIN(3), VI(3), DS(3), XC(3), XSI(3), XSI(3)
9 DIMENSION XOB(3), XDC(3), VTCP(2), BTCP(4), VTCH(2), BTCH(4)
10 DIMENSION VAX(3,3)
11 LOGICAL LSURF, LNPL
12 LOGICAL LSHD, LSTD, LSTS, LCTD, LUCT, LHIT
13 LOGICAL LGHD, LHD, LDEB, LTEST
14 COMMON/TEST/LDEB, LTEST
15 COMMON/PIS/PI, TPI, DPI, RPI
16 COMMON/GEOP/A/X(14,6,3), V(14,6,3), VP(14,6,3), VN(14,3)
17 2, NEP(14), NPI
18 COMMON/EDMAC/VNAG(14,6)
19 COMMON/SCHEFF/XS(3), VXS(3,3)
20 COMMON/INAP/XI(14,14,3), VXI(3,3,14)
21 COMMON/BNDFCL/BD(14,6,2)
22 COMMON/SURFAC/LSURF(14)
23 COMMON/LSHDT/LSHD(14), LSHD(14,14)
24 COMMON/LSHDP/LSTS, LSTD(14)
25 COMMON/FRANG/ANG(14,6)
26 COMMON/HITPL/HPI
27 COMMON/SOURSF/FACTOR
28 COMMON/FAMP/FA, H, HAP
29 COMMON/GROUND/LGHD, NPI
30 LSTS=.FALSE.
31 C!!! SECTION 1 *****
32 C!!! DETERMINATION OF COMMON EDGES
33 C!!! SET NPI=2 FOR ALL EDGES
34 DO 3 NPI=1, NPI
35 NEX=NEP(NPI)
36 DO 3 NE=1, NEX
37 3 FNP(NPI, NE)=2.
38 C!!! STEP THROUGH PLATES (PLATE NP)
39 DO 17 NP=1, NPI
40 NEX=NEP(NP)
41 C!!! STEP THROUGH PLATES (PLATE NO, WHERE NO=NE.NP)
42 DO 16 NO=1, NPI
43 IF(NO.EQ.NP) GO TO 16
44 NFI=NEP(NO)
45 NEC=0
46 NFC=0
47 C!!! STEP AROUND EDGES ON PLATE NP
48 DO 12 NE=1, NEX
49 C!!! STEP AROUND EDGES ON PLATE NO
50 DO 5 NFI=1, NFI
51 XINQ.
52 C!!! IS CORNER NE ON PLATE NP CLOSE TO CORNER NF ON PLATE NO?
53 DO 4 NFI=1, NFI
54 4 XN=XIN(14,NO,NFI)-XINP(NE,NPI)=(XINQ,NFI)-XINP(NE,NPI)
55 IF(XN.LT.0.01) GO TO 6
56 5 CONTINUE
57 GO TO 12
58 6 CONTINUE
59 DO 7 NFI=1, NFI
60 C!!! IF EDGES ARE CLOSE, SET THEM IDENTICAL
61 7 XINQ,NFI=XINP,NE,NPI
62 IF(NEC.NE.0) GO TO 8
63 C!!! CHECK TO SEE IF TWO NEIGHBORING CORNERS ON TWO PLATES
64 C!!! COINCIDE
65 NEC=NE

```

```

00      MFC=MF
01      GO TO 12
02 8     MES=ME-1/EC
03      IF(MES.EQ.1.OR.MES+1.EQ.MEX) GO TO 18
04      GO TO 12
05 18    MEN=MEC
06      IF(MES+1.EQ.MEX) MEN=MEX
07      MFS=IABS(MF-MFC)
08      IF(MFS.EQ.1.OR.MFS+1.EQ.MFX) GO TO 19
09      GO TO 12
10 19    MFN=MFC
11      IF(MFS+1.EQ.MFX) MFN=MFX
12      IF(MF-MFC.EQ.-1) MFN=MF
13      IF(FNP(MP,MEN).GT.0.) GO TO 9
14      NFN=FNP(MP,MEN)*.5
15      NFN=IABS(NFN)/100
16      IF(NFN.EQ.0) GO TO 12
17 5      CONTINUE
18 C!!! STORE WHICH EDGE IS COMMON TO THE EDGE BEING TESTED
19 C!!! FOR LATER USE IN DEFINING WEDGE ANGLES
20      IF(FNP(MO,MFN).GT.0.) FNP(MO,MFN)=-1.*(100-MP+MEN)
21      IF(FNP(MP,MEN).GT.0.) FNP(MP,MEN)=-1.*(100-MO+MFN)
22 12     CONTINUE
23 16     CONTINUE
24 17     CONTINUE
25      IF (LDEBUG) WRITE (6,667)
26 667    FORMAT (/, ' DEBUGGING GEOM SUBROUTINE' )

```

GEOM SECTION 2

PURPOSE

This section computes the edge-fixed coordinate system unit vectors for each edge.

PERTINENT GEOMETRY

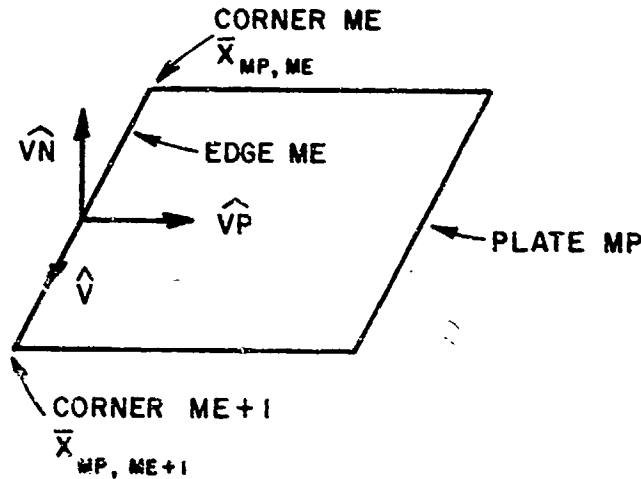


Figure 63--Edge coordinate system unit vectors.

$$\hat{V}_{MP, ME} = \text{edge unit vector} = \hat{x} V(MP, ME, 1) + \hat{y} V(MP, ME, 2) + \hat{z} V(MP, ME, 3)$$

$$\hat{V}_N = \text{plate unit normal} = \hat{x} V_N(MP, 1) + \hat{y} V_N(MP, 2) + \hat{z} V_N(MP, 3)$$

$$\hat{V}_P = \text{edge unit binormal} = \hat{x} V_P(MP, ME, 1) + \hat{y} V_P(MP, ME, 2) + \hat{z} V_P(MP, ME, 3)$$

$$\bar{X}_{MP, ME} = \text{corner location} = \hat{x} X(MP, ME, 1) + \hat{y} X(MP, ME, 2) + \hat{z} X(MP, ME, 3)$$

METHOD

The edge unit vectors are found by,

$$\hat{V}_{MP, ME} = \frac{\bar{X}_{MP, ME+1} - \bar{X}_{MP, ME}}{|\bar{X}_{MP, ME+1} - \bar{X}_{MP, ME}|}$$

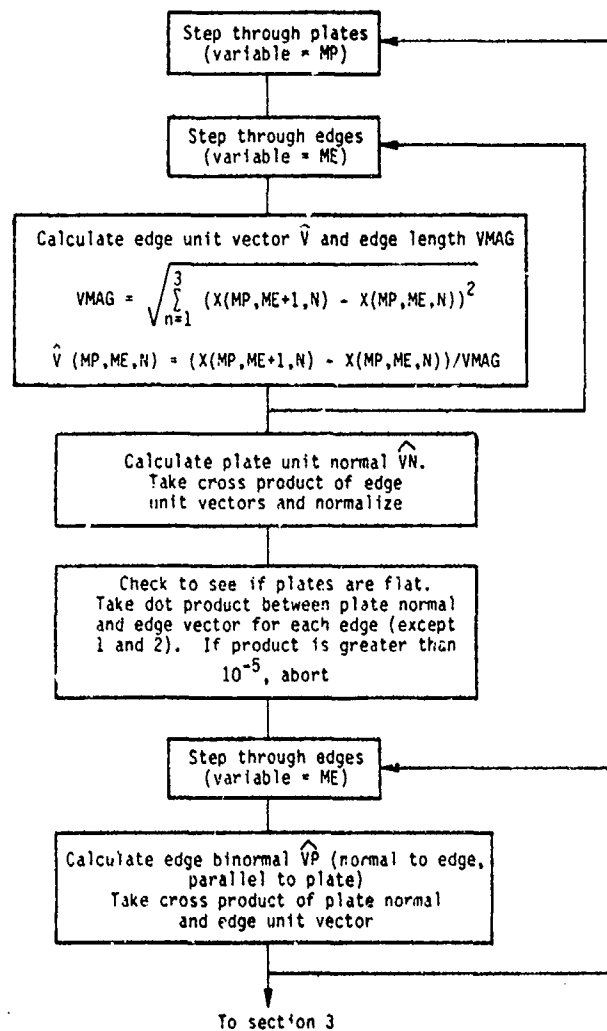
The normals are found using

$$\hat{V}_N = \frac{\sum_{N=1}^{MEX} \hat{V}_{MP, N} \times \hat{V}_{MP, N+1}}{\left| \sum_{N=1}^{MEX} \hat{V}_{MP, N} \times \hat{V}_{MP, N+1} \right|}$$

which is an average over the normals computed by all the edges of the plate. This avoids a possible incorrect normal due to a convex edge geometry. The binormals are found by,

$$\hat{V}_{MP,ME} = \hat{V}_{N_{MP}} \times \hat{V}_{MP,ME}.$$

FLOW DIAGRAM



CODE LISTING

```

93 C!!! SECTION 2 *****
94 C!!! DETERMINATION OF V,VN,AND VP UNIT VECTORS FOR EDGE-FIXED
95 C!!! COORDINATE SYSTEM
96 C!!! STEP THRU PLATES
97 DO 100 MP=1,MPXR
98 MEX=MEX(MP)
99 C!!! STEP THRU EDGES
100 DO 15 ME=1,MEX
101 NME=ME+1
102 IF(NME.GT.MEX) NME=1
103 VM=0.
104 C!!! CALCULATE EDGE UNIT VECTOR V AND EDGE LENGTH VMAG
105 DO 10 N=1,3
106 V(MP,ME,N)=X(MP,NME,N)-X(MP,ME,N)
107 VM=VM+V(MP,ME,N)*V(MP,ME,N)
108 VMAG(MP,ME)=SQRT(VM)
109 DO 11 N=1,3
110 V(MP,ME,N)=V(MP,ME,N)/VMAG(MP,ME)
111 CONTINUE
112 IF(.NOT.LDEBUG)GO TO 991
113 DO 992 ME=1,MEX
114 WRITE(6,*) (V(MP,ME,N),N=1,3)
115 992 CONTINUE
116 991 CONTINUE
117 C!!! CALCULATE PLATE UNIT NORMAL VN
118 VN(MP,1)=0.
119 VN(MP,2)=0.
120 VN(MP,3)=0.
121 DO 22 ME=1,MEX
122 MV=ME+1
123 IF(MV.GT.MEX) MV=1
124 VN(MP,1)=VN(MP,1)+V(MP,ME,2)*V(MP,MV,3)-V(MP,MV,2)*V(MP,ME,3)
125 VN(MP,2)=VN(MP,2)+V(MP,ME,3)*V(MP,MV,1)-V(MP,MV,3)*V(MP,ME,1)
126 VN(MP,3)=VN(MP,3)+V(MP,ME,1)*V(MP,MV,2)-V(MP,MV,1)*V(MP,ME,2)
127 22 CONTINUE
128 VNM=0.
129 DO 20 N=1,3
130 VNM=VNM+VN(MP,N)*VN(MP,N)
131 VNM=SQRT(VNM)
132 DO 21 N=1,3
133 VN(MP,N)=VN(MP,N)/VNM
134 IF (LDEBUG) WRITE (6,*) (VN(MP,N),N=1,3)
135 C!!! INSURE THAT ALL PLATES ARE FLAT. OTHERWISE ABORT!
136 C!!! TAKE DOT PRODUCT OF PLATE NORMAL AND EACH EDGE UNIT VECTOR
137 DO 120 ME=3,MEX
138 DOT=VN(MP,1)*V(MP,ME,1)+VN(MP,2)*V(MP,ME,2)+VN(MP,3)*V(MP,ME,3)
139 IF(ABS(DOT).LT.1.E-3)GO TO 120
140 MEE=ME+1
141 WRITE(6,121)MP,MEE
142 121 FORMAT(' PLATE # ',12,' IS NOT FLAT! CORNER # ',12,' HAS ')
143 2/PROBLEM. PROGRAM ABORTS! *****
144 STOP
145 120 CONTINUE
146 C!!! CALCULATE UNIT BINORMAL VP WHICH IS IN PLATE PLANE
147 C!!! AND PERPENDICULAR TO PLATE EDGE
148 C!!! TAKE CROSS PRODUCT OF PLATE NORMAL AND EDGE VECTOR
149 DO 30 ME=1,MEX
150 VP(MP,ME,1)=VN(MP,2)*V(MP,ME,3)-VN(MP,3)*V(MP,ME,2)
151 VP(MP,ME,2)=VN(MP,3)*V(MP,ME,1)-VN(MP,1)*V(MP,ME,3)
152 30 VP(MP,ME,3)=VN(MP,1)*V(MP,ME,2)-VN(MP,2)*V(MP,ME,1)
153 IF(.NOT.LDEBUG)GO TO 993
154 DO 994 ME=1,MEX
155 994 WRITE(6,*) (VP(MP,ME,N),N=1,3)
156 993 CONTINUE
157 120 CONTINUE

```

GEOM SECTION 3

PURPOSE

To calculate source image information for reflection from plates.

PERTINENT GEOMETRY

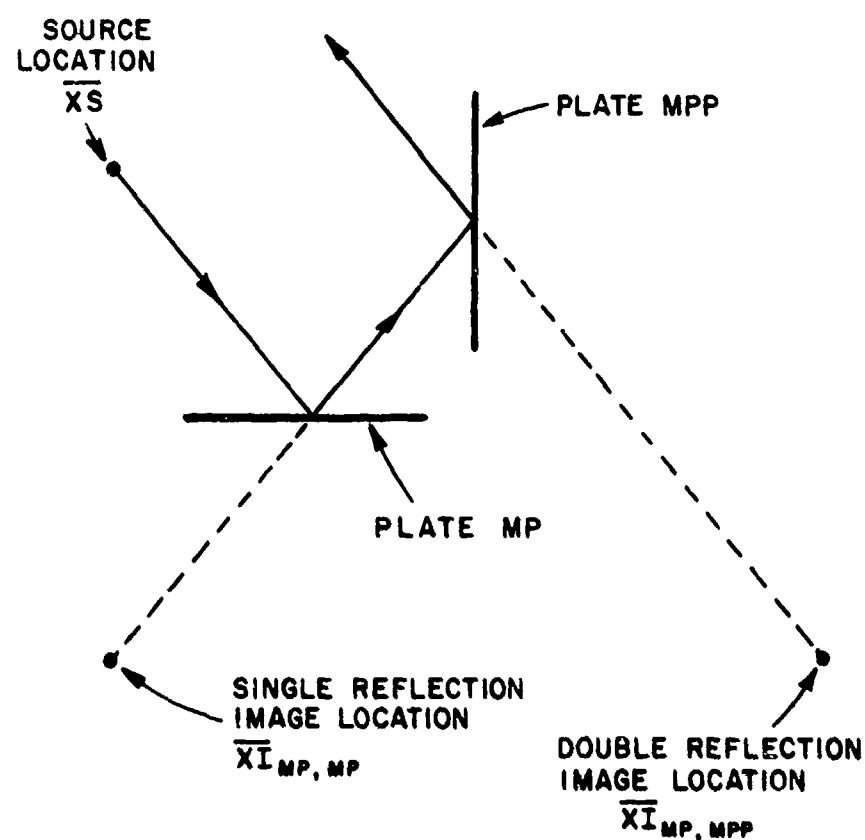
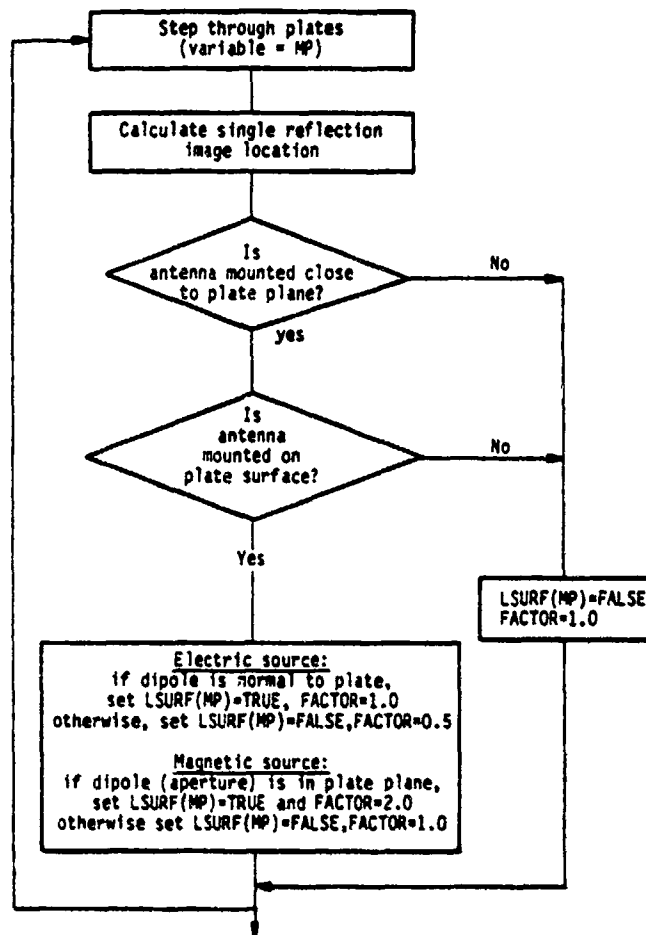


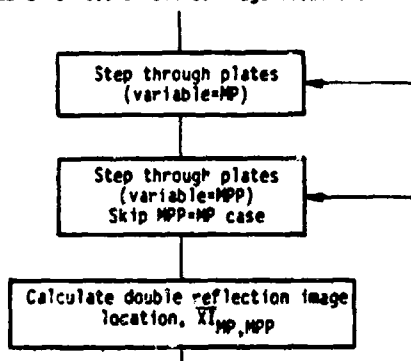
Figure 64--Geometry of image locations for a doubly-reflected ray.

FLOW DIAGRAM

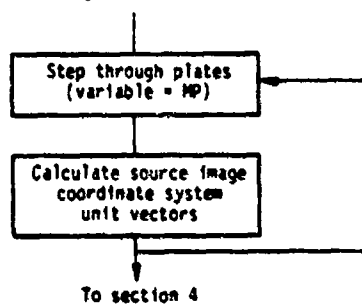
1. Determination of single reflection source image locations and the constant, FACTOR



2. Calculate double reflection source image locations



3. Determination of single reflection image dipole directions (image of the source coordinate system axes unit vectors).



```

158 C111 SECTION 3 *****
159 C111 DETERMINATION OF SOURCE IMAGE INFORMATION FOR SINGLE
160 C111 AND DOUBLE REFLECTION FROM PLATES
161 C111 1. DETERMINATION OF SINGLE REFLECTION SOURCE IMAGE LOCATIONS
162 C111 AND THE CONSTANT, FACTOR, FOR SOURCES MOUNTED ON THE PLATE
163 C111 SURFACES
164 FACTOR=1.
165 C111 STEP THRU PLATES
166 DO 50 MP=1,MPXR
167 LSURF(MP)=.FALSE.
168 C111 CALCULATE SINGLE REFLECTION IMAGE LOCATION
169 CALL IMAGE(XII,XS,AN,MP)
170 C111 IS ANTENNA MOUNTED ON PLATE PLANE?
171 IF(ABS(AN).GT.1.E-5)GO TO 560
172 C111 MOVE SOURCE LOCATION SLIGHTLY OFF PLATE IN DIRECTION
173 C111 OF PLATE NORMAL
174 DO 566 N=1,3
175 566 XSI(N)=XS(N)+1.E-5*VN(MP,N)
176 CALL IMAGE(XSII,XSI,AN,MP)
177 DSM=0.
178 DO 563 N=1,3
179 DS(N)=XSI(N)-XSII(N)
180 563 DSM=DSM+DS(N)*DS(N)
181 DSM=SQRT(DSM)
182 DO 564 N=1,3
183 XIN(N)=XSII(N)
184 564 DS(N)=DS(N)/DSM
185 CALL PLAIN(XIN,DS,DHIT,-MP,LHIT)
186 IF(.NOT.LHIT)GO TO 560
187 DO 567 N=1,3
188 XS(N)=XSI(N)
189 567 XII(N)=XSII(N)
190 ENORM=VN(MP,1)*VXS(3,1)+VN(MP,2)*VYS(3,2)+VN(MP,3)*VXS(3,3)
191 IF(ENORM.EQ.0)GO TO 561
192 C111 IS MONOPOLE NORMAL TO PLATE?
193 IF(1.-ABS(ENORM).GT.1.E-3)GO TO 562
194 LSURF(MP)=.TRUE.
195 GO TO 560
196 562 FACTOR=0.5
197 GO TO 560
198 C111 IS SLOT IN PLATE PLANE?
199 561 IF(ABS(ENORM).GT.1.E-3)GO TO 560
200 LSURF(MP)=.TRUE.
201 FACTOR=2.
202 560 DO 50 N=1,3
203 50 XI(MP,MP,N)=XII(N)
204 C111 2. CALCULATE DOUBLE REFLECTION SOURCE IMAGE LOCATIONS
205 DO 55 MP=1,MPXR
206 DO 55 MPP=1,MPXR
207 IF(MP.EQ.MPP)GO TO 55
208 DO 51 N=1,3
209 51 XIN(N)=XI(MP,MP,N)
210 CALL IMAGE(XII,XIN,AN,MPP)
211 DO 52 N=1,3
212 52 XI(MP,MPP,N)=XII(N)
213 55 CONTINUE
214 IF (LDEBUG) WRITE (6,*) (((XI(MP,MPP,N),N=1,3),MPP=1,MPXR),
215 2MP=1,MPXR)
216 C111 3. DETERMINATION OF SINGLE REFLECTION IMAGE DIPOLE
217 C111 DIRECTIONS
218 DO 57 MP=1,MPXR
219 CALL INDIR(VAX,VXS,MP)
220 DO 57 NI=1,3
221 DO 57 NJ=1,3
222 57 VXI(NI,NJ,MP)=VAX(NI,NJ)
223 IF(.NOT.LDEBUG)GO TO 551
224 DO 552 MP=1,MPXR
225 DO 552 NI=1,3
226 552 WRITE(6,*) MP,NI,(VXI(NI,NJ,MP),NJ=1,3)
227 551 CONTINUE

```

GEOM SECTION 4

PURPOSE

To determine permissible range for angle β_0 for diffraction of source ray off of plate edge.

PERTINENT GEOMETRY

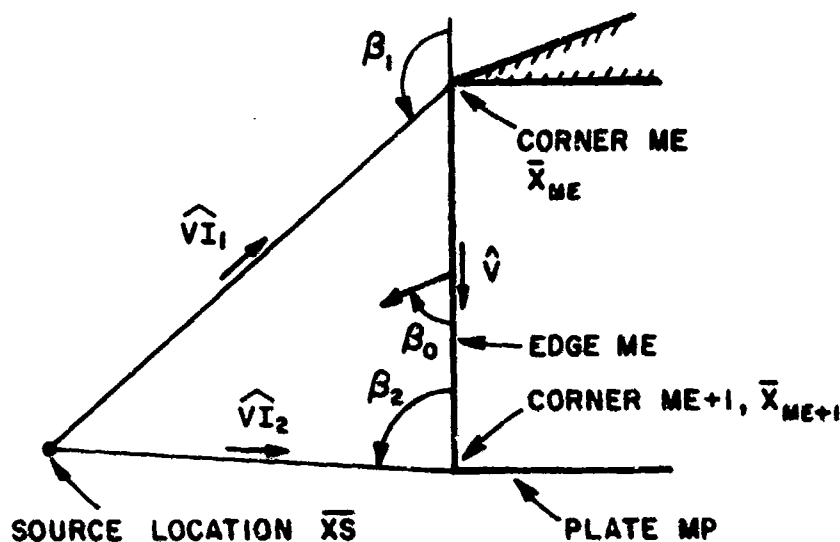


Figure 65--Geometry for determining diffraction angle range.

METHOD

The law of diffraction dictates that diffraction from a plate edge is possible when

$$\cos\beta_1 \leq \cos\beta_0 \leq \cos\beta_2.$$

where β_0 is the angle that the incident and diffracted rays make with the edge (see Figure 65). β_1 and β_2 are diffraction angle limits and are defined in terms of their cosines as:

$$BD(MP, ME, 1) = \cos\beta_1 = \hat{VI}_1 \cdot \hat{V}$$

$$BD(MP, ME, 2) = \cos\beta_2 = \hat{VI}_2 \cdot \hat{V}.$$

where

$$\hat{V}_{I_1} = \frac{\bar{x}_{ME} - \bar{x}_S}{|\bar{x}_{ME} - \bar{x}_S|}$$

$$\hat{V}_{I_2} = \frac{\bar{x}_{ME+1} - \bar{x}_S}{|\bar{x}_{ME+1} - \bar{x}_S|}.$$

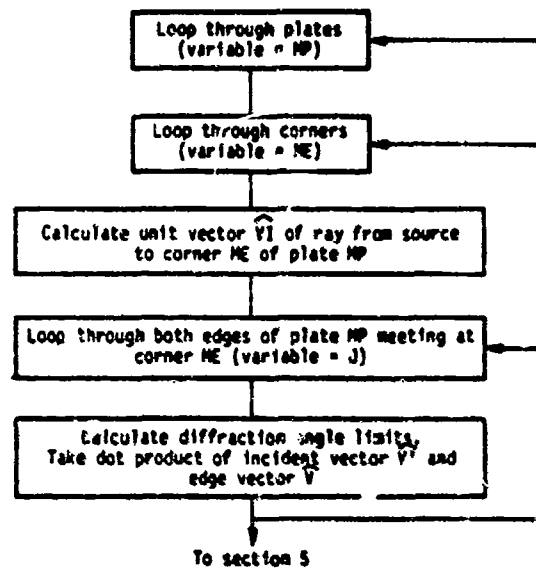
The vectors mentioned above relate to the code as follows:

$$\bar{x}_{ME} = \hat{x} X(MP, ME, 1) + \hat{y} X(MP, ME, 2) + \hat{z} X(MP, ME, 3)$$

$$\bar{x}_S = \hat{x} X_S(1) + \hat{y} X_S(2) + \hat{z} X_S(3)$$

$$\hat{V} = \hat{x} V(MP, ME, 1) + \hat{y} V(MP, ME, 2) + \hat{z} V(MP, ME, 3).$$

FLOW DIAGRAM



CODE LISTING

```

228 C!!! SECTION 4 *** *****
229 C!!! DETERMINATION OF PERMISSABLE RANGE FOR DIFFRACTION ANGLE
230 C!!! LOOP THRU PLATES
231      DO 42 MP=1,MPXR
232      MEX=MEP(MP)
233 C!!! LOOP THRU CORNERS
234      DO 41 ME=1,MEX
235      VIN=0.
236 C!!! CALCULATE VECTOR VI FROM SOURCE TO CORNER ME OF PLATE MP
237      DO 40 N=1,3
238      VI(N)=X(MP,ME,N)-XS(N)
239 40    VIN=VIN+VI(N)*VI(N)
240      VIN=SQRT(VIN)
241 C!!! LOOP THRU BOTH EDGES MEETING AT CORNER ME
242      DO 41 J=1,2
243      MJ=ME+1-J
244      IF(MJ.EQ.0) MJ=MEX
245      BD(MP,MJ,J)=0.
246 C!!! CALCULATE BD, THE DOT PRODUCT OF INCIDENT RAY
247 C!!! VECTOR VI AND EDGE VECTOR V
248      DO 41 N=1,3
249 41    BD(MP,MJ,J)=BD(MP,MJ,J)+V(MP,MJ,N)*VI(N)/VIN
250 42    CONTINUE
251      IF(.NOT.LDEBUG)GO TO 995
252      DO 996 MP=1,MPX
253      MEX=MEP(MP)
254      DO 996 ME=1,MEX
255 996    WRITE(6,*) (BD(MP,ME,J),J=1,2)
256 995    CONTINUE

```

GEOM SECTION 5

PURPOSE

To calculate wedge angles for plates with common edges.

PERTINENT GEOMETRY

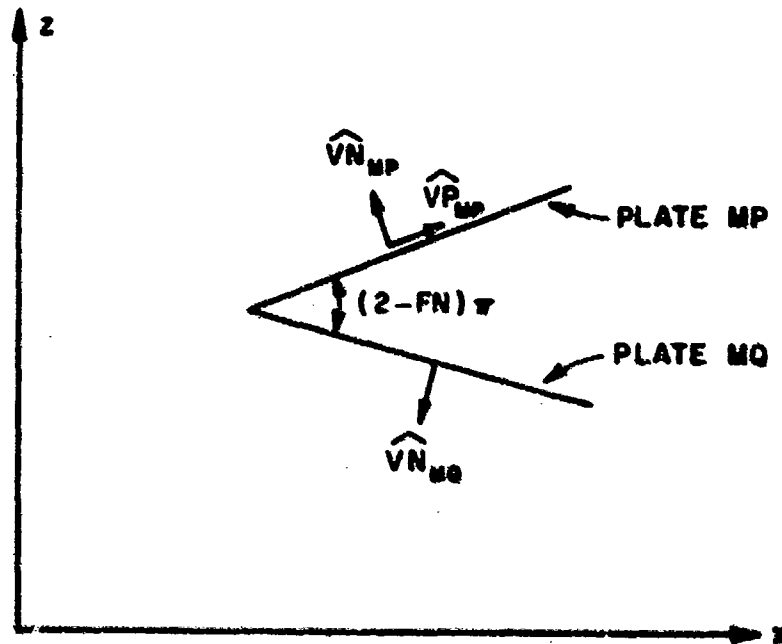


Figure 66--Geometry used to determine wedge angles of plates with common edges.

METHOD

The wedge angle is specified using the wedge angle number FN, such that the wedge angle is given by

$$(2 - FN)\pi$$

as shown in Figure 66. The wedge angle number is determined as follows:

$$FN = \frac{1}{\pi} \tan^{-1} \left(\frac{TOP}{BOT} \right).$$

where

$$TOP = \hat{V}N_{MQ} \cdot \hat{V}P_{MP}$$

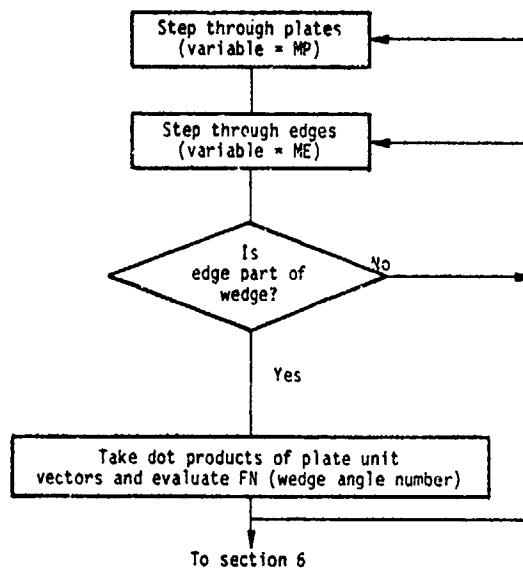
$$BOT = -\hat{V}N_{MP} \cdot \hat{V}N_{MQ}$$

$$\hat{V}N_{MP} = \hat{x} VN(MP,1) + \hat{y} VN(MP,2) + \hat{z} VN(MP,2)$$

$$\hat{V}P_{MP} = \hat{x} VP(MP,ME,1) + \hat{y} VP(MP,ME,2) + \hat{z} VP(MP,ME,3)$$

$$\hat{V}N_{MQ} = \hat{x} VN(MQ,1) + \hat{y} VN(MQ,2) + \hat{z} VN(MQ,3).$$

FLOW DIAGRAM



CODE LISTING

```

257 C!!! SECTION 5 *****
258 C!!! DETERMINATION OF WEDGE ANGLES FOR PLATES WITH COMMON EDGES
259 C!!! STEP THROUGH PLATES
260 DO 35 MP=1,MPX
261 MEX=MEX(MP)
262 C!!! STEP THROUGH EDGES
263 DO 35 ME=1,MEX
264 C!!! IS EDGE ME PART OF A WEDGE?
265 IF(FNP(MP,ME).GT.-5.) GO TO 35
266 NFN=FNP(MP,ME)-.5
267 NFN=IABS(NFN)
268 MQ=NFN/100
269 MF=NFN-MQ*100
270 IF(FNP(MQ,MF).GT.0.) GO TO 35
271 C!!! TAKE DOT PRODUCTS OF PLATE UNIT VECTORS AND EVALUATE
272 C!!! FN (WEDGE ANGLE NUMBER)
273 BOT=-(VN(MP,1)*VN(MQ,1)+VN(MP,2)*VN(MQ,2)+VN(MP,3)*VN(MQ,3))
274 TOP=VP(MP,ME,1)*VN(MQ,1)+VP(MP,ME,2)*VN(MQ,2)+
275 2VP(MP,ME,3)*VN(MQ,3)
276 FANG=BTAN2(TOP,BOT)
277 ANN=0.
278 ANP=0.
279 DO 34 N=1,3
280 XSX=XS(N)-X(MP,ME,N)
281 ANN=ANN+XSX*VN(MP,N)
282 34 ANP=ANP+XSX*VP(MP,ME,N)
283 PHWAR=BTAN2(ANN,ANP)
284 IF(PHWAR.LT.0.) PHWAR=TPH+PHWAR
285 FN=FANG/PI
286 IF(PHWAR.GT.FN*PI) FN=2.-ARS(FANG)/PI
287 FNP(MP,ME)=FN
288 35 CONTINUE
289 IF(.NOT.LDERUG)GO TO 997
290 DO 998 MP=1,MPX
291 MEX=MEX(MP)
292 DO 998 ME=1,MEX
293 998 WRITE(6,*)FNP(MP,ME)
294 997 CONTINUE

```

GEOM SECTION 6

PURPOSE

To determine plates which are totally shadowed from the source.

PERTINENT GEOMETRY

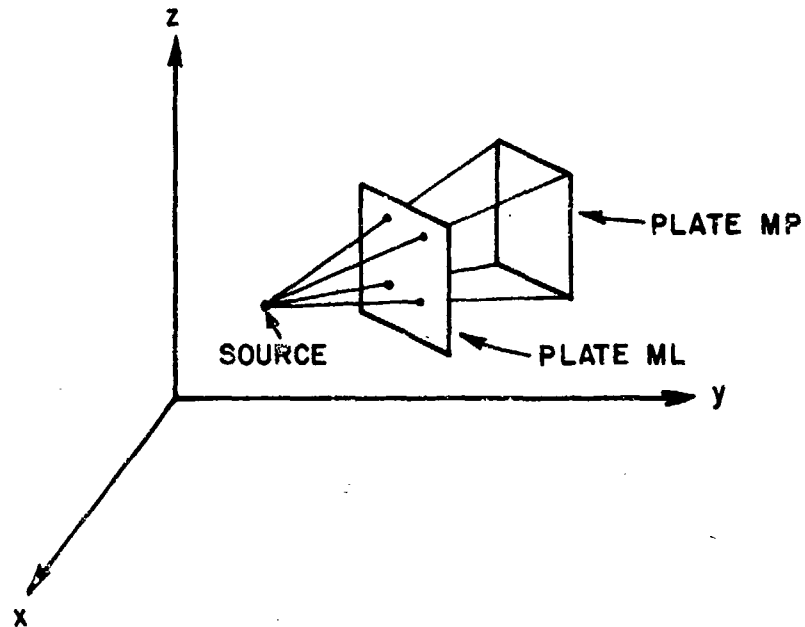


Figure 67a--Configuration where plate ML totally shadows plate MP from source.

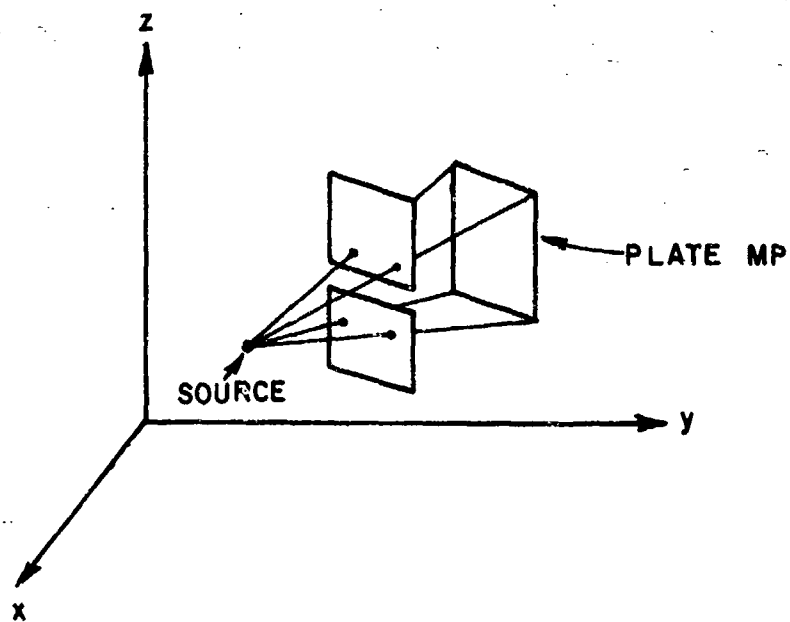
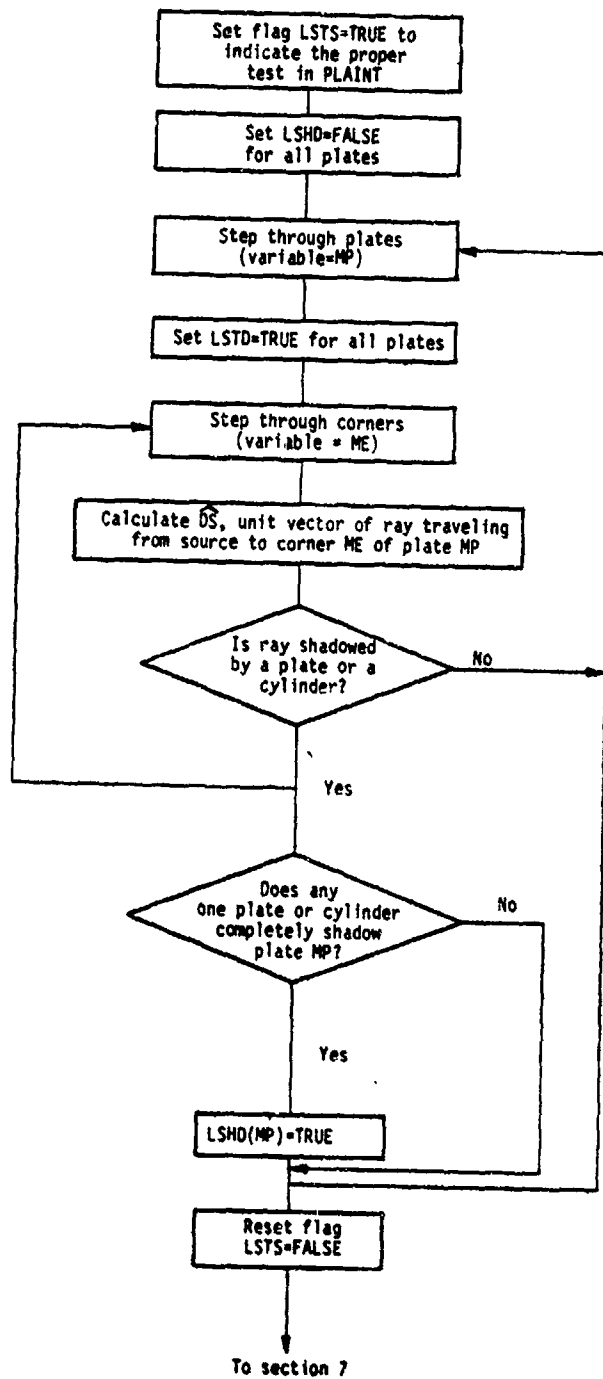


Figure 67b--Configuration where plate MP is not totally shadowed from the source

METHOD

If plate ML totally shadows plate MP from the source, then every ray drawn from the source to a corner of plate MP will intersect plate ML. The routine computes vectors from the source to each corner of plate MP and uses a shadow testing algorithm to check if any plate shadows all of the rays (see Figures 67a and 67b). If so, it is assumed that plate MP is totally shadowed from the source.

FLOW DIAGRAM



```

295 C!!! SECTION 6 *****
296 C!!! DETERMINATION OF PLATES THAT ARE TOTALLY
297 C!!! SHADOWED FROM THE SOURCE
298 LSTS=.TRUE.
299 DO 72 MP=1,MPXR
300 72 LSHD(MP)=.FALSE.
301 C!!! STEP THRU PLATES
302 DO 79 MP=1,MPX
303 MEX=MEP(MP)
304 C!!! SET LSTD=TRUE FOR ALL PLATES
305 C!!! SET LCTD=TRUE FOR THE CYLINDER
306 DO 73 ML=1,MPX
307 73 LSTD(ML)=.TRUE.
308 LCTD=.TRUE.
309 C!!! STEP THRU CORNERS
310 DO 77 ME=1,MEX
311 DSM=0.
312 C!!! CALCULATE DS, UNIT VECTOR OF RAY TRAVELING FROM SOURCE TO
313 C!!! CORNER ME OF PLATE MP
314 DO 74 N=1,3
315 DS(N)=X(MP,ME,N)-XS(N)
316 74 DSM=DSM+DS(N)*DS(N)
317 DSM=SQRT(DSM)
318 DO 75 N=1,3
319 75 DS(N)=DS(N)/DSM
320 C!!! IS RAY SHADOWED BY PLATE OR CYLINDER?
321 CALL PLAIN(XS,DS,DHIT,MP,LHIT)
322 IF(LHIT.AND.DHIT.GT.DSM)LHIT=.FALSE.
323 IF(.NOT.LCTD) GO TO 76
324 PHCR=BTAN2(DS(2),DS(1))
325 CALL CYLINT(XS,DS,PHCR,DHIT,LHCT,.FALSE.)
326 IF(.NOT.LHCT) LCTD=.FALSE.
327 IF(LHCT.AND.DHIT.GT.DSM) LCTD=.FALSE.
328 76 CONTINUE
329 IF(.NOT.LHIT.AND..NOT.LCTD) GO TO 79
330 // CONTINUE
331 C!!! CHECK TO SEE IF ANY ONE PLATE ML COMPLETELY SHADOWS PLATE MP
332 C!!! STEP THRU PLATES
333 DO 78 ML=1,MPX
334 IF(.NOT.LSTD(ML)) GO TO 78
335 LSHD(MP)=.TRUE.
336 78 CONTINUE
337 IF(LCTD) LSHD(MP)=.TRUE.
338 79 CONTINUE
339 IF (LDEBUG) WRITE (6,*) (LSHD(MP),MP=1,MPXR)
340 LSTS=.FALSE.

```

GEOM SECTION 7

PURPOSE

This section handles various calculations for plates which intersect each other.

PERTINENT GEOMETRY

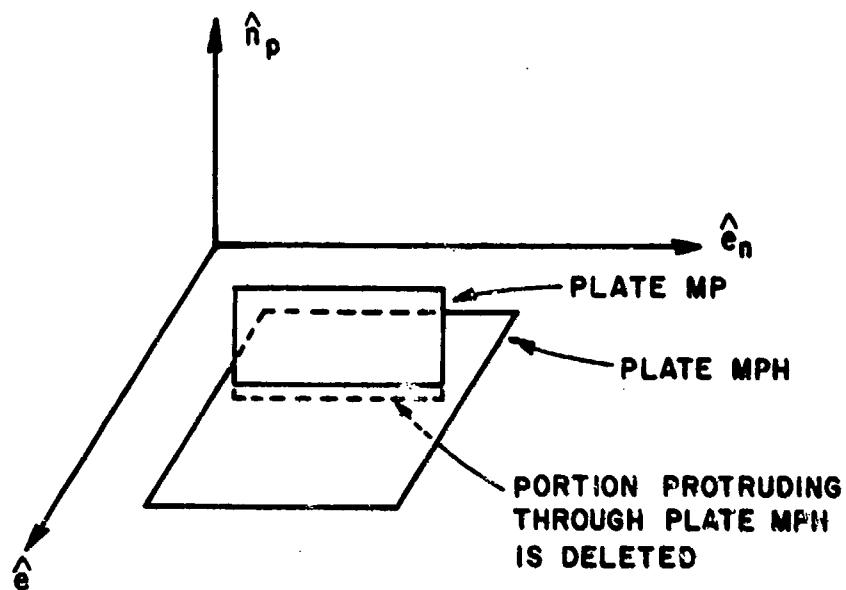
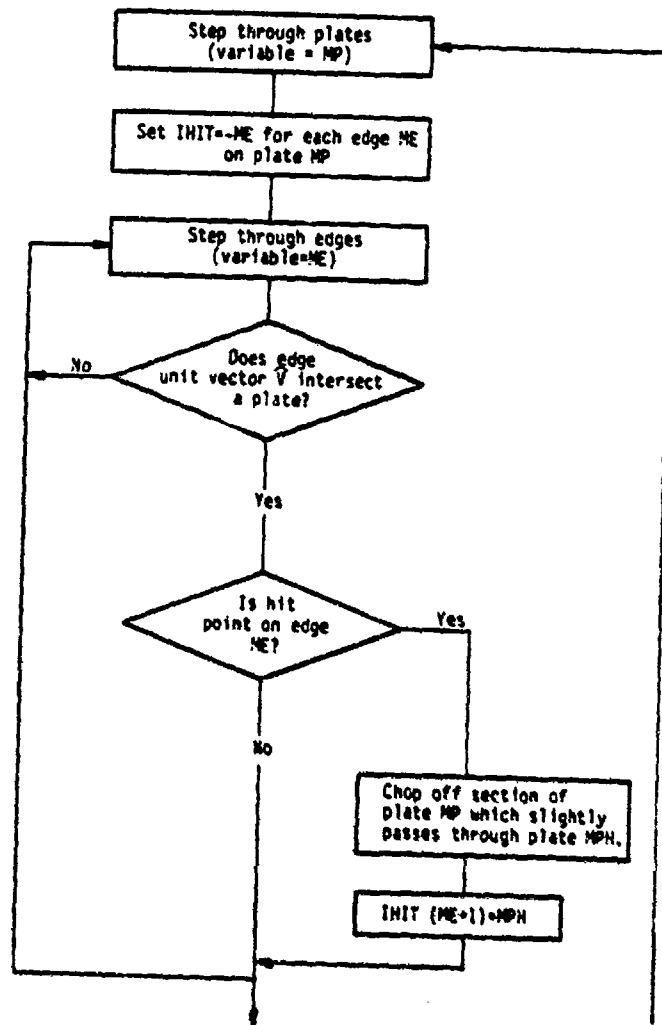


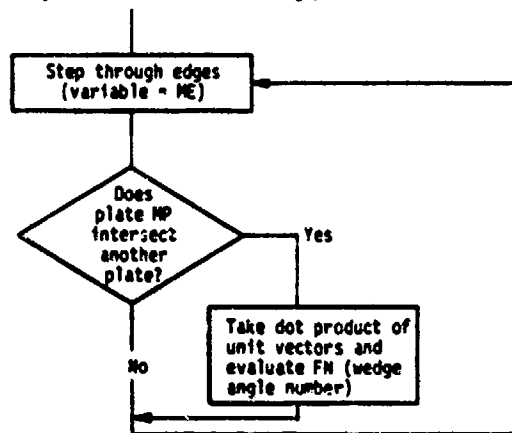
Figure 68--Illustration of a plate which intersects another.

FLOW DIAGRAM

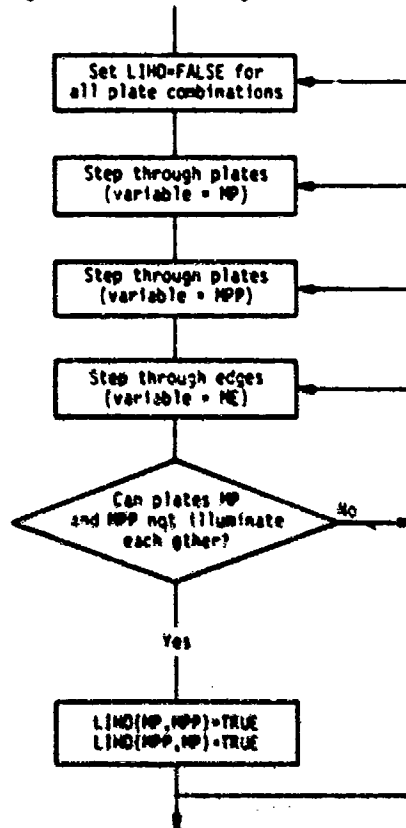
1. Determine plates that intersect



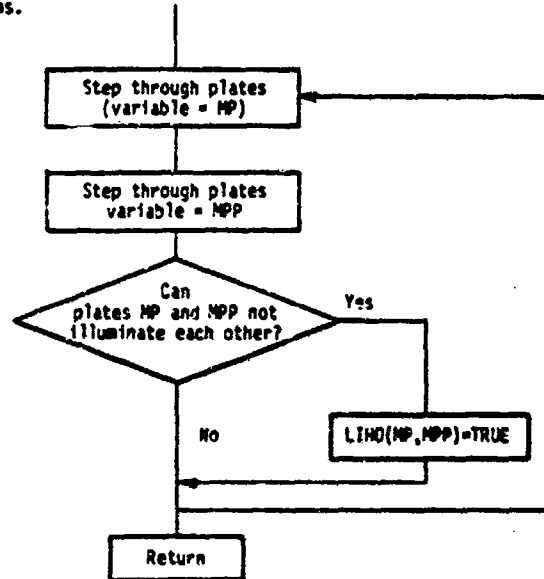
2. Determine new wedge angle number for intersecting plates



3. Find plates with common edges which cannot illuminate each other because wedge angle between plates is greater than 180° .



4. Determine which plates cannot illuminate each other based on illegal image locations.



CODE LISTING

```

341 C!!! SECTION 7 *****
342 C!!! 1. DETERMINE PLATES THAT INTERSECT
343 C!!! STEP THRU PLATES
344 DO 85 MP=1,MPX
345 MEX=MEX(MP)
346 C!!! SET IHIT=ME FOR EACH EDGE ON PLATE MP
347 DO 889 ME=1,MEX
348 889 IHIT(ME)=ME
349 C!!! STEP THRU EDGES
350 DO 82 ME=1,MEX
351 DO 81 N=1,3
352 XIN(N)=X(MP,ME,N)
353 81 DS(N)=V(MP,ME,N)
354 C!!! DOES EDGE INTERSECT ANOTHER PLATE?
355 CALL PLAIN(XIN,DS,IHIT,MP,LHIT)
356 IF(.NOT.LHIT)GO TO 80
357 IF(DHIT.GT.VMAG(MP,ME))GO TO 80
358 MC=ME+1
359 IF(MC.GT.MEX)MC=1
360 C!!! CHOP OFF SECTOR OF PLATE MP WHICH PASSES THRU PLATE MPH
361 IF(DHIT.LT.0.1)GO TO 83
362 DO 82 N=1,3
363 82 X(MP,MC,N)=X(MP,ME,N)+(DHIT-2.E-5)*V(MP,ME,N)
364 VMAG(MP,ME)=DHIT-2.E-5
365 IHIT(MC)=MPH
366 IF(LDEBUG)WRITE(6,*)MP,MC,(X(MP,MC,N),N=1,3)
367 GO TO 80
368 83 VMAG(MP,ME)=VMAG(MP,ME)-DHIT
369 DO 84 N=1,3
370 84 X(MP,ME,N)=X(MP,MC,N)-VMAG(MP,ME)*V(MP,ME,N)
371 IHIT(ME)=MPH
372 IF(LDEBUG)WRITE(6,*)MP,ME,(X(MP,ME,N),N=1,3)
373 80 CONTINUE
374 C!!! 2. DETERMINE NEW WEDGE ANGLE NUMBER FOR INTERSECTING PLATES
375 C!!! STEP THRU EDGES
376 DO 80 ME=1,MEX
377 MC=ME+1
378 IF(MC.GT.MEX)MC=1
379 C!!! DO PLATES INTERSECT?
380 IF(IHIT(ME).NE.IHIT(MC))GO TO 86
381 MH=IHIT(ME)
382 C!!! TAKE DOT PRODUCTS OF PLATE UNIT VECTORS AND EVALUATE FN
383 XX=VN(MH,1)*VP(MP,ME,1)+VN(MH,2)*VP(MP,ME,2)+VN(MH,3)
384 2*VP(MP,ME,3)
385 YY=VN(MH,1)*VN(MP,1)+VN(MH,2)*VN(MP,2)+VN(MH,3)*VN(MP,3)
386 FN=MH
387 IF(XX.LE.0.)GO TO 89
388 FN=0.5*BTAN2(YY,XX)/P
389 ANN=0.
390 DO 87 N=1,3
391 87 ANN=ANN+VN(MP,N)*(XS(N)-X(MP,ME,N))
392 IF(ANN.GT.0.)GO TO 88
393 FN=3.-FN
394 GO TO 88
395 89 WRITE(6,*)MP,MR
396 881 FORMAT(' HAVING PLATES ',215,' INTERSECT YET GEOMETRY '
397 3.' INDICATES ATTACHED PLATE IS SHADOWED!!!!',//)
398 88 FNP(MP,ME)=FN
399 80 CONTINUE
400 85 CONTINUE
401 IF(.NOT.LDEBUG)GO TO 887
402 DO 888 MP=1,MPX
403 MEX=MEX(MP)
404 DO 888 ME=1,MEX
405 888 WRITE(6,*)FNP(MP,ME)
406 887 CONTINUE

```

```

407 C111 3. DETERMINE PLATES WITH COMMON EDGES WHICH CANNOT ILLUMINATE
408 C111 EACH OTHER.
409 C111 SET LIHD=FALSE FOR ALL PLATE COMBINATIONS
410      DO 90 MP=1,MPXR
411      DO 90 MPP=1,MPXR
412 90      LIHD(MP,MPP)=.FALSE.
413 C111 STEP THRU PLATES
414      DO 91 MP=1,MPX
415 C111 STEP THRU PLATES
416      DO 91 MPP=1,MPX
417      MEX=MEX(MPP)
418 C111 STEP THRU EDGES
419      DO 92 ME=1,MEX
420 C111 CAN PLATES MP AND MPP NOT ILLUMINATE EACH OTHER?
421 C111 IF SO, IDENTIFY
422      IFN=FNP(MPP,ME)/100.
423      IF(IFN.ME.MP)GO TO 92
424      MEH=FNP(MPP,ME)-IFN*100*0.5
425      IF(FNP(IFN,MEH).LT.1.)GO TO 92
426      LIHD(MP,MPP)=.TRUE.
427      LIHD(MPP,MP)=.TRUE.
428 92      CONTINUE
429 91      CONTINUE
430 C111 4. DETERMINE PLATES WHICH CANNOT ILLUMINATE EACH OTHER BASED
431 C111 ON ILLEGAL IMAGE LOCATIONS.
432 C111 STEP THRU PLATES
433      DO 921 MP=1,MPX
434      MEX=MEX(MP)
435      SUMT=.E30
436      DO 922 ME=1,MEX
437      SUM=.0.
438      DO 923 N=1,3
439 923      SUM=SUM+(X(MP,ME,N)-X(M1,N))**2
440      IF(SUM.GT.SUMT)GO TO 922
441      SUMT=SUM
442      MEE=ME
443 922      CONTINUE
444      DO 924 MPP=1,MPX
445      ANP=.0.
446      ANI=.0.
447      DO 925 N=1,3
448      ANP=ANP+(X(MP,MEE,N)-X(MPP,1,N))**VN(MPP,N)
449 925      ANI=ANI+(X(MP,MPP,N)-X(MPP,1,N))**VN(MPP,N)
450 C111 CAN PLATES MP AND MPP NOT ILLUMINATE EACH OTHER?
451      IF(ANP*ANI.LT.0.)GO TO 924
452      LIHD(MP,MPP)=.TRUE.
453 924      CONTINUE
454 921      CONTINUE
455      IF(.NOT.DEBUG)GO TO 93
456      DO 94 MP=1,MPX
457      DO 94 MPP=1,MPX
458 94      WRITE(6,*)MP,MPP,LIHD(MP,MPP)
459 93      CONTINUE
460      RETURN
461      END

```

SYMBOL DICTIONARY

AN	DOT PRODUCT OF PLATE UNIT NORMAL AND VECTOR FROM SOURCE TO THE PLATE (CALCULATED IN IMAGE)
ANI	DOT PRODUCT OF VECTOR FROM CORNER 1 OF PLATE MPP TO THE DOUBLE REFLECTION IMAGE LOCATION XI(MP,MPP) AND THE UNIT NORMAL OF PLATE MPP
ANN	DOT PRODUCT OF XSX AND UNIT NORMAL OF PLATE NP
ANP	DOT PRODUCT OF VECTOR FROM CORNER 1 OF PLATE MPP TO CORNER MEE OF PLATE NP AND UNIT NORMAL OF PLATE MPP ALSO DOT PRODUCT OF XSX AND BINORMAL OF EDGE ME OF PLATE NP
BU	COSINES OF ANGLES DEFINING BOUNDS ON DIFFRACTION ANGLE
DOT	NEGATIVE DOT PRODUCT OF UNIT NORMALS OF PLATES NP AND NO
DNIT	DISTANCE FROM SOURCE TO NEAREST HIT (FROM PLAIN)
DOT	DOT PRODUCT OF PLATE UNIT NORMAL AND EDGE UNIT NORMAL
DS	UNIT VECTOR OF RAY FROM SOURCE TO CORNER ME OF PLATE NP (SECTION 6) ALSO UNIT VECTOR OF RAY FROM IMAGE TO SOURCE (SECTION 3) ALSO UNIT VECTOR OF EDGE ME (SECTION 7)
DSM	NORMALIZATION CONSTANT FOR DS
ENDNM	DOT PRODUCT OF VN (THE UNIT NORMAL OF PLATE NP) AND THE Z AXIS OF THE SOURCE COORDINATE SYSTEM
FACTOR	MAGNITUDE ADJUSTMENT FOR SOURCES MOUNTED ON THE SURFACE OF PLATES
FANG	EDGE ANGLE
FN	EDGE ANGLE NUMBER
FNP	EDGE ANGLE NUMBER (ALSO USED IN DEFINING COMMON EDGES)
IFA	INDEX VARIABLE
INI	STORES PLATE NUMBERS FOR PLATES INTERSECTED BY AN EDGE
J	DO LOOP VARIABLE
LCN	SET TRUE IF RAY HITS CYLINDER (RETURNED FROM CYLINT)
LCID	SET TRUE IF CYLINDER SHADOWS PLATE FROM SOURCE
LNIT	SET TRUE IF RAY INTERSECTS A PLATE (FROM PLAIN)
LINO	SET TRUE IF PLATES NP AND NPP CANNOT ILLUMINATE EACH OTHER
LNOO	SET TRUE IF PLATE NP IS TOTALLY SHADOWED FROM THE SOURCE
LSID	SET TRUE IF PLATE NL TOTALLY SHADOWS PLATE NP FROM THE SOURCE
LSTE	SET TRUE IF TOTAL SHADOWING ALGORITHM IS BEING USED
ML	INDEX VARIABLE
ME	DO LOOP VARIABLE, ALSO INDEX VARIABLE
MEL	WORKING VARIABLE
MEE	INDEX VARIABLE
MEN	INDEX VARIABLE
MEX	WORKING VARIABLE
MES	COMPUTATIONAL VARIABLE
MFA	NUMBER OF EDGES ON A GIVEN PLATE
MF	DO LOOP VARIABLE, ALSO INDEX VARIABLE
MFC	WORKING VARIABLE
MFN	WORKING VARIABLE
MFS	COMPUTATIONAL VARIABLE
MFA	NUMBER OF EDGES ON PLATE NO
NL	DO LOOP VARIABLE
NM	INDEX VARIABLE
NP	DO LOOP VARIABLE (STEP THRU PLATES) ALSO PLATE INDEX VARIABLE

MPP DO LOOP VARIABLE (ALSO PLATE INDEX VARIABLE)
 MO DO LOOP VARIABLE (STEP THRU PLATES).
 ALSO INDEX VARIABLE
 MH INDEX VARIABLE
 MV INDEX VARIABLE
 N DO LOOP VARIABLE
 NPH WORKING VARIABLE
 NI DO LOOP VARIABLE
 NJ DO LOOP VARIABLE
 PHCN PHI COMPONENT OF VECTOR FROM SOURCE TO PLATE
 CORNER IN RCS
 PHCAN ANGLE WHICH DETERMINES WHICH SIDE OF THE
 INTERSECTING PLATES IS ILLUMINATED
 SQA LENGTH OF VECTOR FROM SOURCE TO EDGE ME
 OF PLATE MP
 SUMT LENGTH OF VECTOR FROM SOURCE TO CLOSEST EDGE
 OF PLATE MP
 TOP DOT PRODUCT OF BINORMAL OF COMMON EDGE OF PLATE MP
 AND NORMAL OF PLATE MO
 V MATRIX OF X,Y,Z COMPONENTS DEFINING EDGE UNIT
 VECTORS IN RCS
 VAX X,Y,Z COMPONENTS DEFINING SINGLE REFLECTION
 IMAGE SOURCE COORDINATE SYSTEM AXES IN RCS
 COMPONENTS
 VI X,Y,Z COMPONENTS OF UNIT VECTOR OF RAY FROM SOURCE
 TO CORNER ME OF PLATE MP
 VIA NORMALIZATION CONSTANT OF VI
 VM DISTANCE BETWEEN TWO NEIGHBORING CORNERS ON A
 PLATE SQUARED
 VMAG DISTANCES BETWEEN NEIGHBORING CORNERS ON PLATES
 VN X,Y,Z COMPONENTS DEFINING PLATE UNIT NORMAL
 DIRECTIONS IN RCS COMPONENTS
 VNA PLATE UNIT NORMAL NORMALIZATION CONSTANT
 VP MATRIX CONTAINING EDGE UNIT BINORMAL DIRECTIONS IN
 REFERENCE COORDINATE SYSTEM
 VXi X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF SOURCE
 IMAGE COORDINATE SYSTEM AXES IN RCS
 VxS X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF SOURCE
 COORDINATE SYSTEM AXES IN RCS
 XI ARRAY OF COMPONENTS DEFINING SINGLE AND DOUBLE
 REFLECTION SOURCE IMAGE LOCATIONS IN RCS
 XII X,Y,Z COMPONENTS OF SINGLE REFLECTION IMAGE
 LOCATION CALCULATED IN SUBROUTINE IMAGE
 XIN X,Y,Z COMPONENTS OF LOCATION OF CORNER ME OF PLATE
 MP
 ALSO, SINGLE REFLECTION IMAGE LOCATION
 ALSO, XII
 XB DISTANCE BETWEEN CORNER ME ON PLATE MO AND CORNER
 ME ON PLATE MP
 XSI X,Y,Z COMPONENTS OF SOURCE LOCATION MOVED A SMALL
 AMOUNT IN THE DIRECTION OF THE PLATE NORMAL
 FROM SOURCE ON PLATE PLANE
 XSII X,Y,Z COMPONENTS OF SOURCE IMAGE LOCATION
 CALCULATED IN SUBROUTINE IMAGE FOR SOURCE
 LOCATED AT XSI
 XSA X,Y,Z COMPONENTS OF VECTOR FROM CORNER ME OF PLATE
 MP TO THE SOURCE
 XX DOT PRODUCT OF BINORMAL OF EDGE ME OF PLATE MP AND
 NORMAL OF PLATE MO
 YY DOT PRODUCT OF NORMALS OF PLATES MP AND MO

GEOMC

PURPOSE

To calculate geometry associated with fixed cylinder structures (end cap normals, etc.).

PERTINENT GEOMETRY

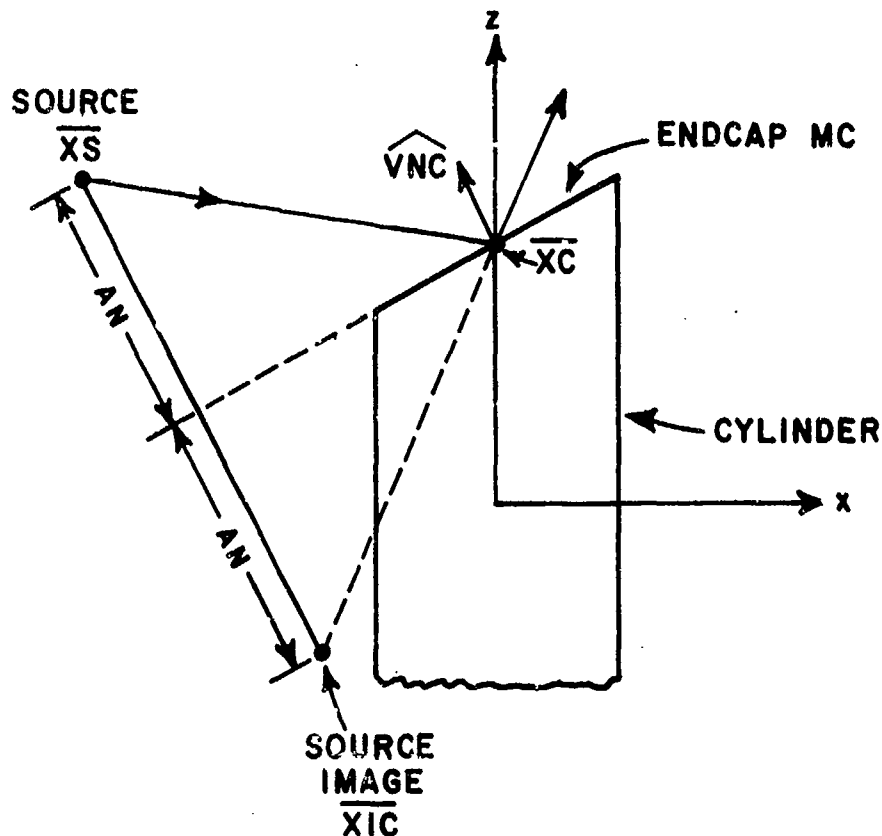


Figure 69-- Geometry for determining source image location for reflection from cylinder end cap.

$$\hat{VNC} = \hat{x} VNC(1) + \hat{y} VNC(2) + \hat{z} VNC(3)$$

$$\overline{XS} = \hat{x} XS(1) + \hat{y} XS(2) + \hat{z} XS(3)$$

$$\overline{XIC} = \hat{x} XIC(MC,1) + \hat{y} XIC(MC,2) + \hat{z} XIC(MC,3)$$

$$\overline{XC} = \hat{z} ZC(MC)$$

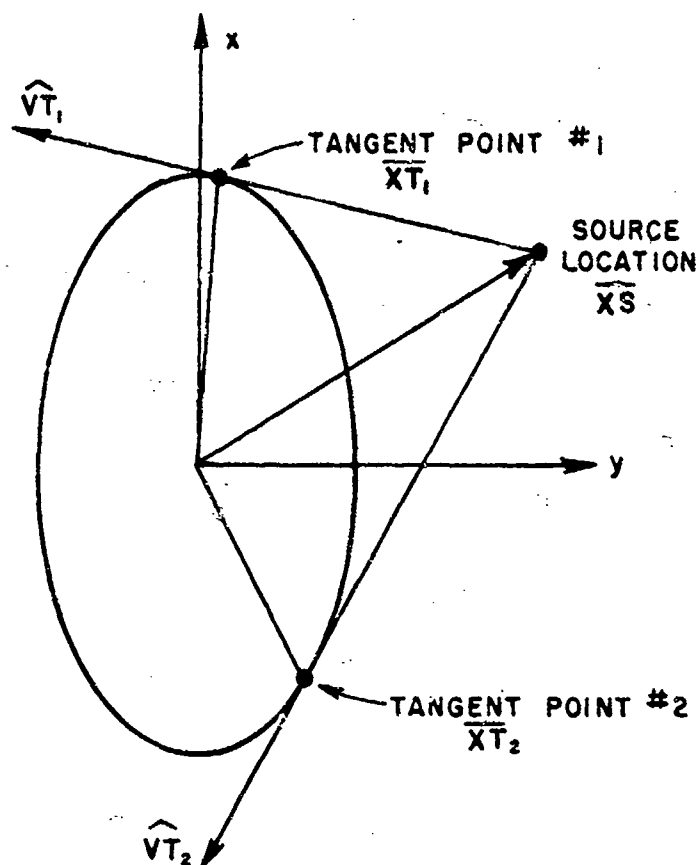


Figure 70-- Illustration of vectors from the source tangent to the elliptic cylinder.

$$\hat{V}_{T_1} = \hat{x} \text{ BTS}(1) + \hat{y} \text{ BTS}(2)$$

$$\hat{V}_{T_2} = \hat{x} \text{ BTS}(3) + \hat{y} \text{ BTS}(4)$$

$$\overline{X}_{T_1} = \hat{x} A \cos(\text{VTS}(1)) + \hat{y} B \sin(\text{VTS}(1))$$

$$\overline{X}_{T_2} = \hat{x} A \cos(\text{VTS}(2)) + \hat{y} B \sin(\text{VTS}(2))$$

$$\overline{X}_S = \hat{x} \text{ XS}(1) + \hat{y} \text{ XS}(2) + \hat{z} \text{ XS}(3)$$

METHOD

The image source location is given by:

$$\overline{X}_{IC} = \overline{X}_S - 2 \text{ AN } \hat{V}_{NC},$$

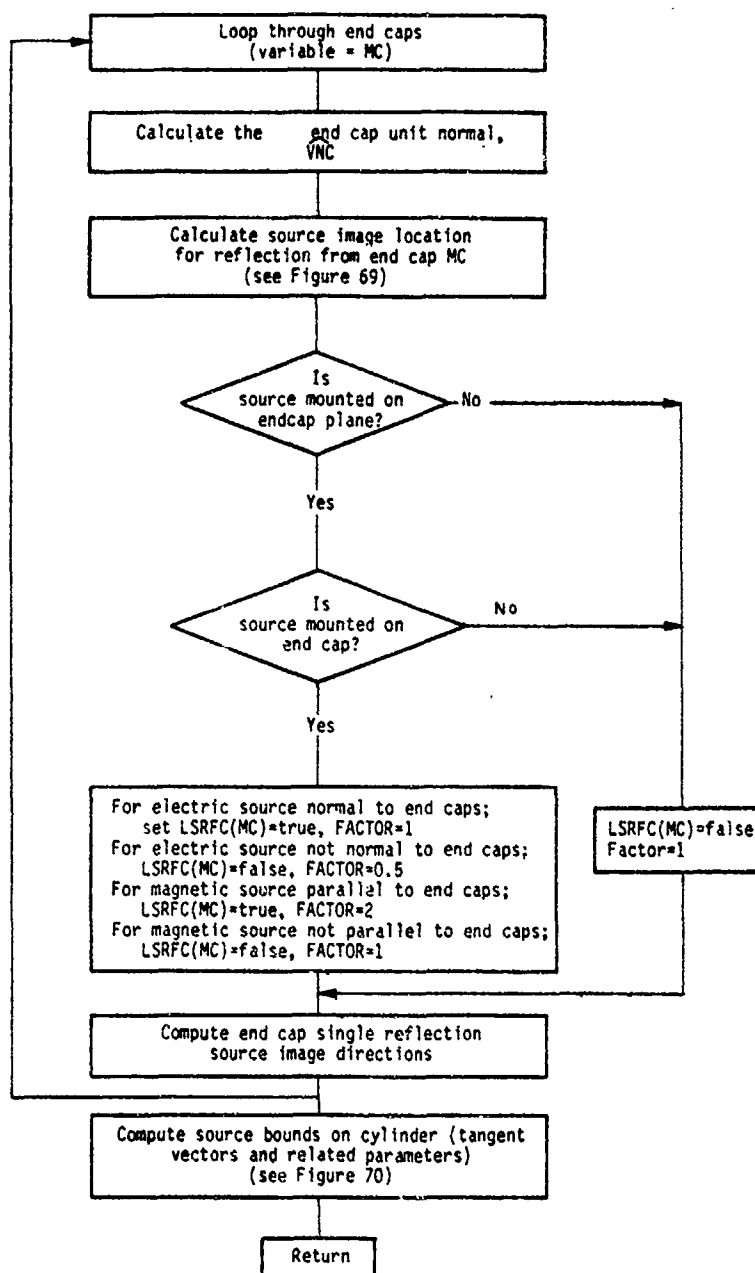
where

$$AN = (\overline{XS} - \overline{XC}) \cdot \hat{VNC}.$$

This is illustrated in Figure 69.

The tangent vectors from the source to the cylinder, as illustrated in Figure 70, are found in subroutine TANG.

FLOW DIAGRAM



SYMBOL DICTIONARY

AN	DOT PRODUCT OF END CAP NORMAL AND RAY FROM END CAP TO SOURCE
DS	UNIT VECTOR OF RAY FROM SOURCE IMAGE TO SOURCE
DSM	DISTANCE FROM SOURCE IMAGE TO SOURCE
ENORM	DOT PRODUCT OF END CAP NORMAL AND Z AXIS OF SOURCE COORDINATE SYSTEM
LHIT	SET TRUE IF RAY HITS END CAP (FROM SUB. CAPINT)
LSRFC	SET TRUE IF SOURCE IS MOUNTED ON END CAP MC
MC	END CAP INDEX VARIABLE
N	DO LOOP VARIABLE
NC	SIGN CHANGE VARIABLE
NI	DO LOOP VARIABLE
NJ	DO LOOP VARIABLE
VNC	X,Y, AND Z COMPONENTS OF THE END CAP UNIT NORMAL IN REF COORD SYS
VXIC	X,Y,Z COMPONENTS OF UNIT VECTORS DEFINING AXES OF END CAP SOURCE IMAGE COORDINATE SYSTEM
XIN	SOURCE IMAGE LOCATION IN END CAP MC

CODE LISTING

```

1 C-----
2      SUBROUTINE GEOMC
3 C!!!
4 C!!! THIS ROUTINE COMPUTES ALL THE GEOMETRY ASSOCIATED
5 C!!! WITH FIXED CYLINDER STRUCTURES, END CAP NORMALS, ETC.
6 C!!!
7      DIMENSION XIN(3),DS(3),VNC(3),VAX(3,3)
8      LOGICAL LPLA,LCYL,LSRFC,LHIT,LDEBUG,LTEST
9      COMMON/PIS/PI,TPI,DPR,RPD
10     COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
11     COMMON/SORINF/XS(3),VXS(3,3)
12     COMMON/IMCINF/XIC(2,3),VXIC(3,3,2)
13     COMMON/FARP/IM,H,HAW
14     COMMON/SOURSF/FACTOR
15     COMMON/ENDSCL/DTS,VTS(2),BTS(4)
16     COMMON/SRFACC/LSRFC(2)
17     COMMON/LPLCY/LPLA,LCYL
18     COMMON/TEST/LDEBUG,LTEST
19     IF(LDEBUG) WRITE(6,900)
20 900   FORMAT(/,' DEBUGGING GEOMC SUBROUTINE')
21 C!!! DETERMINATION OF DISK IMAGES
22     IF(.NOT.LPLA) FACTOR=1.
23 C!!! LOOP THRU END CAPS
24     DO 515 MC=1,2
25     LSRFC(MC)=.FALSE.
26     NC=MC
27     IF(MC.EQ.2) NC=-1
28 C!!! CALCULATE END CAP UNIT NORMAL
29     VNC(1)=-NC*CNC(MC)
30     VNC(2)=0.
31     VNC(3)=NC*SNC(MC)
32 C!!! CALCULATE SOURCE IMAGE LOCATION FOR REFLECTION FROM
33 C!!! END CAP MC
34     AN=XS(1)*VNC(1)+XS(2)*VNC(2)+(XS(3)-ZC(MC))*VNC(3)
35     DO 510 N=1,3
36 510   XIC(MC,N)=XS(N)-2.*AN*VNC(N)
37 C!!! IS SOURCE MOUNTED ON END CAP PLANE?
38     IF(ABS(AN).GT.1.E-5) GO TO 520
39     DO 526 N=1,3
40 526   XS(N)=XS(N)+1.E-5*VNC(N)
41     AN=XS(1)*VNC(1)+XS(2)*VNC(2)+(XS(3)-ZC(MC))*VNC(3)
42     DO 527 N=1,3
43 527   XIC(MC,N)=XS(N)-2.*AN*VNC(N)
44 C!!! IS ANTENNA MOUNTED ON END CAP, IF SO IDENTIFY
45     DSM=0.
46     DO 523 N=1,3
47     DS(N)=XS(N)-XIC(MC,N)
48 523   DSM=DSM+DS(N)*DS(N)
49     DSM=SQRT(DSM)
50     DO 524 N=1,3
51     DS(N)=DS(N)/DSM
52 524   XIN(N)=XIC(MC,N)
53     CALL CAPINT(XIN,DS,DHIT,MC,LHIT)
54     IF(.NOT.LHIT) GO TO 520
55     ENORM=VNC(1)*VXS(3,1)+VNC(2)*VXS(3,2)+VNC(3)*VXS(3,3)
56     IF(IM.NE.0) GO TO 521
57     IF(1.-ABS(ENORM).GT.1.E-3) GO TO 522
58     LSRFC(MC)=.TRUE.
59     GO TO 520
60 522   FACTOR=.5
61     GO TO 520
62 521   IF(ABS(ENORM).GT.1.E-3) GO TO 520
63     LSRFC(MC)=.TRUE.
64     FACTOR=2.
65 520   CONTINUE

```

```

66 C!!! COMPUTE END CAP IMAGE SOURCE AXES DIRECTIONS
67 CALL IMCDIR(VAX,VXS,VNC)
68 DO 530 NJ=1,3
69 DO 530 NI=1,3
70 530 VXIC(NI,NJ,MC)=VAX(NI,NJ)
71 515 CONTINUE
72 IF(.NOT.LDEBUG) GO TO 910
73 DO 911 MC=1,2
74 WRITE(6,*) MC,LSRFC(MC)
75 WRITE(6,*) (XIC(MC,N),N=1,3)
76 DO 912 NI=1,3
77 912 WRITE(6,*) NI,(VXIC(NI,NJ,MC),NJ=1,3)
78 911 CONTINUE
79 910 CONTINUE
80 C!!! DETERMINATION OF SOURCE BOUNDS ON CYLINDER
81 CALL TANG(DTS,VTS,BTS,XS)
82 IF(.NOT.LDEBUG) GO TO 915
83 WRITE(6,*) DTS
84 WRITE(6,*) VTS(1),VTS(2)
85 WRITE(6,*) (BTS(J),J=1,4)
86 915 CONTINUE
87 RETURN
88 END

```

GEOMPC

PURPOSE

To compute variables pertaining to plate-cylinder interactions which are constant for a given set of plates and cylinder and a given source.

PERTINENT GEOMETRY

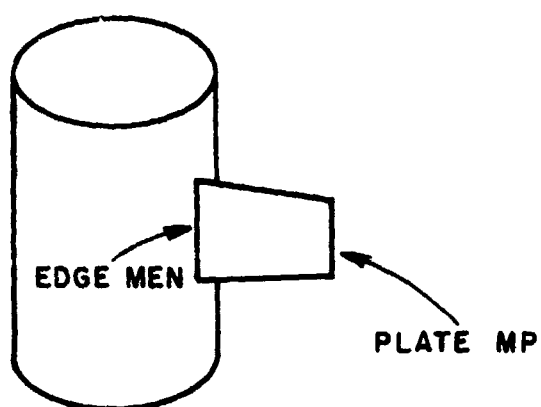


Figure 71-- Illustration of plate attached to cylinder as detailed in section 1.

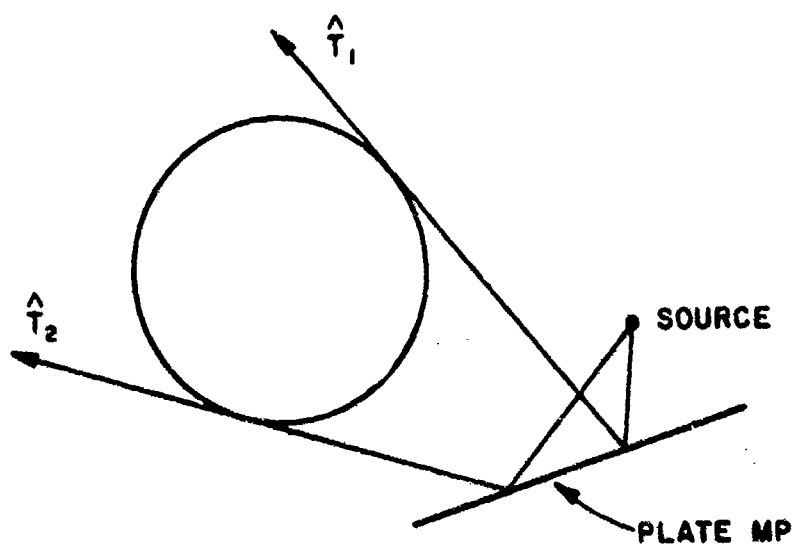


Figure 72-- Illustration of source rays reflected by plate MP tangent to the cylinder as detailed in section 2.

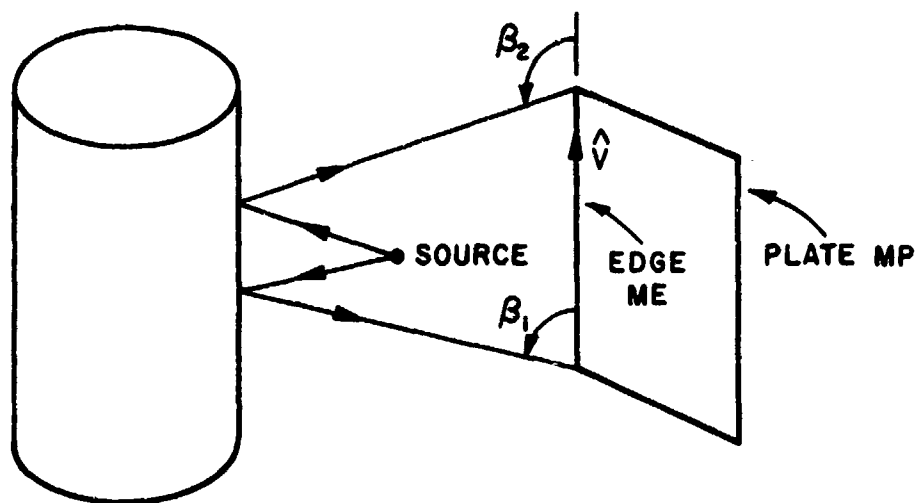


Figure 73-- Illustration of bounds for cylinder reflected, plate diffracted region detailed in section 3.

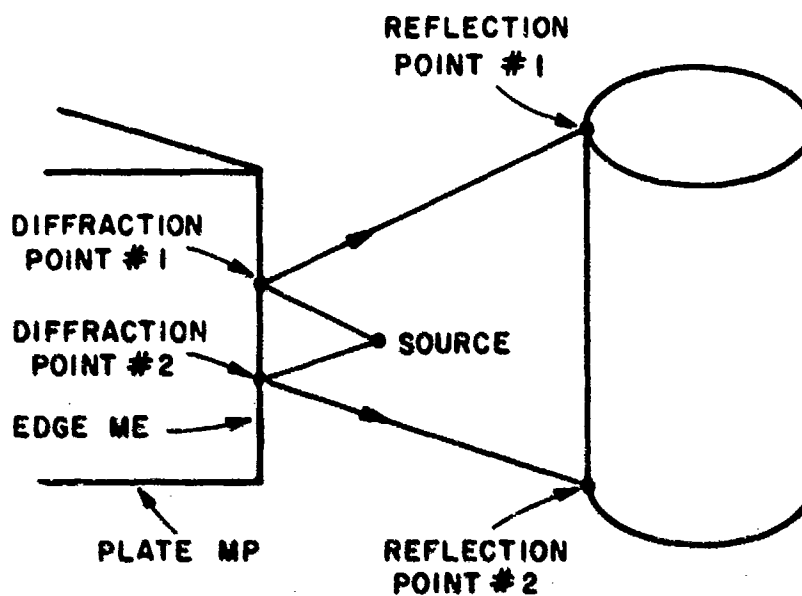


Figure 74-- Illustration of starting point path for plate diffracted, cylinder reflected ray tracing algorithm as detailed in section 4.

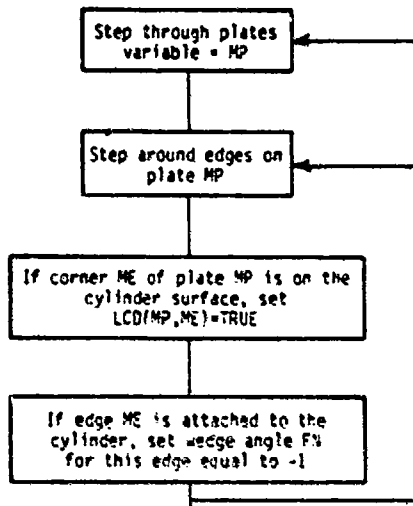
METHOD

The bounds for cylinder reflected, plate diffracted fields are illustrated in Figure 73. Details of the method used to find these parameters are given on pages 149-154 of Reference 1. Also see the write-up for subroutine RFDPT. The starting point paths for plate diffracted, cylinder reflected fields are illustrated in Figure 74. Details of the method used to find these parameters are given on pages 161-163 of Reference 1. Also see the write-up for subroutine DFRPT.

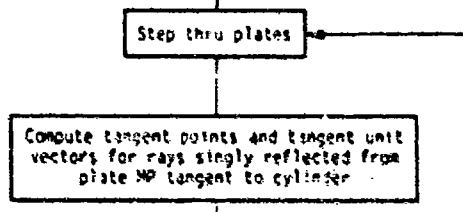
FLOW DIAGRAM

FLOW DIAGRAM

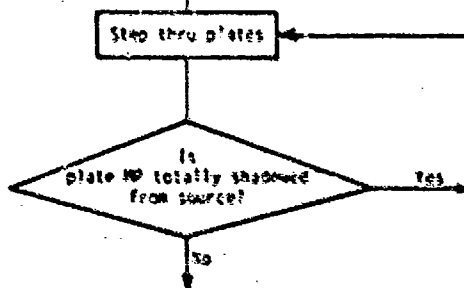
1. Determine corners and edges which are attached to the cylinder.

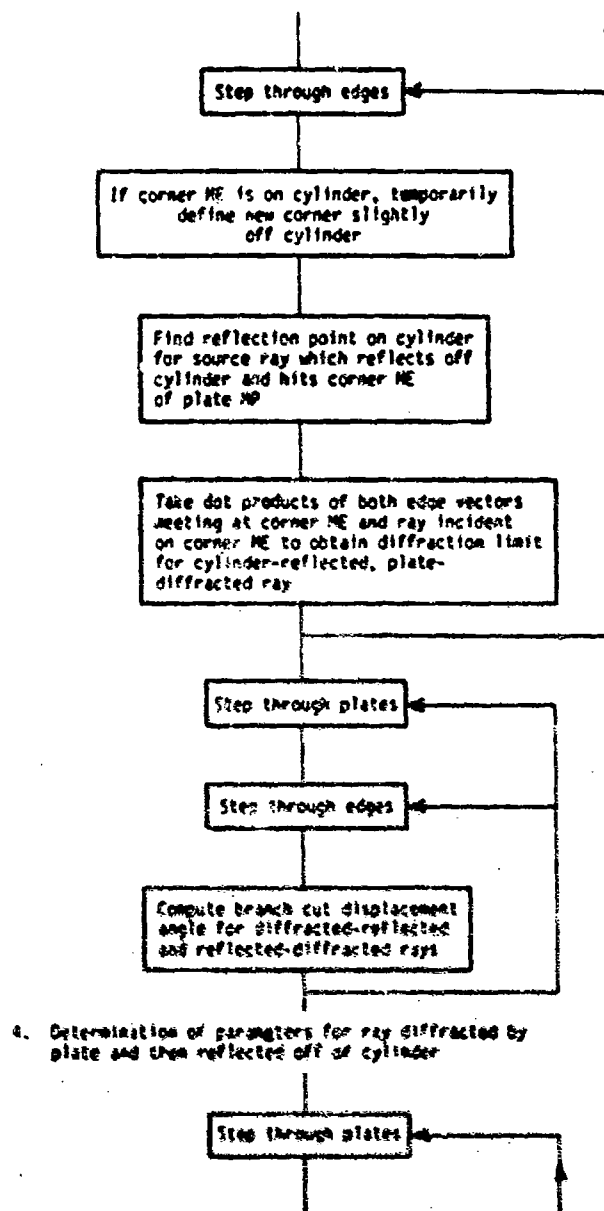


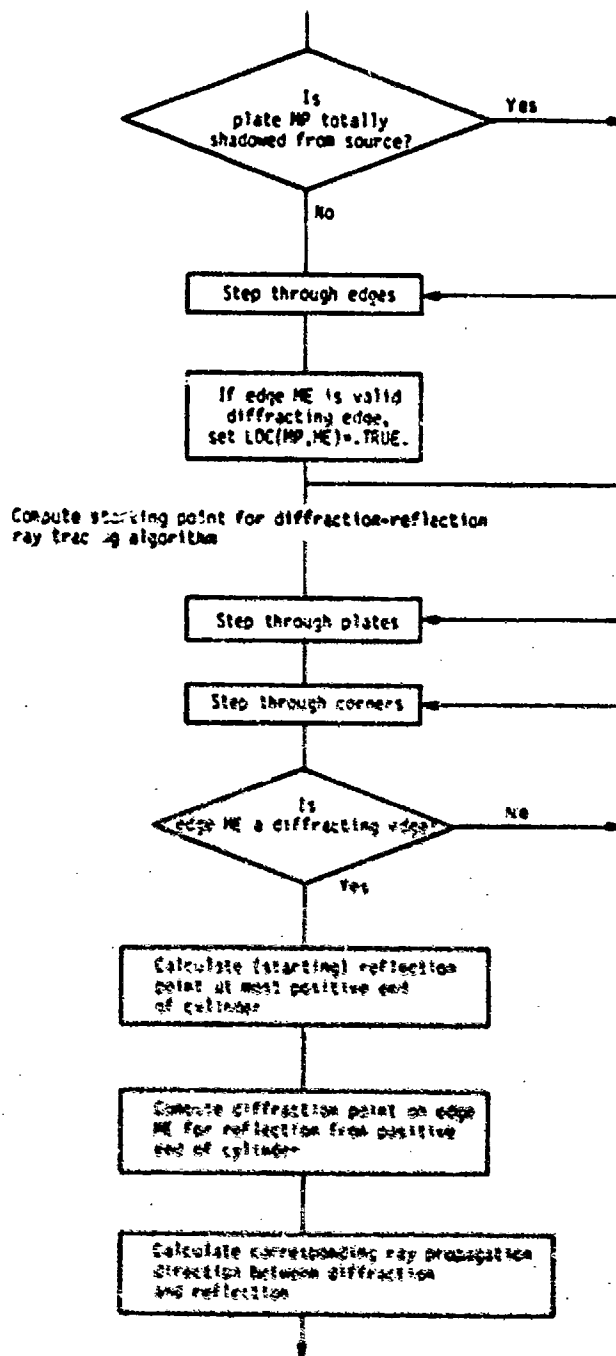
2. Determination of image bounds on cylinder

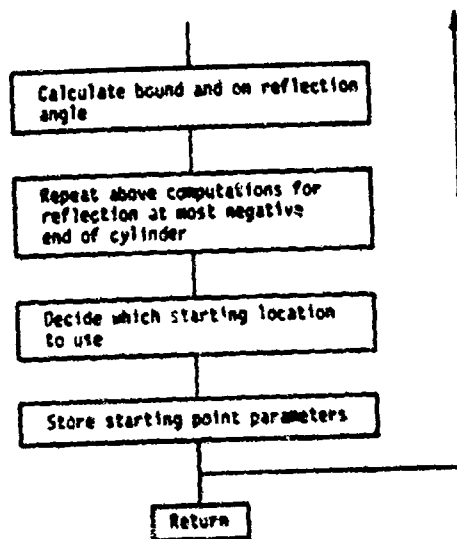


3. Determination of permissible range for cylinder reflected, plate diffracted term









SYMBOL DICTIONARY

BCD	DIFFRACTION LIMITS FOR RAY REFLECTED BY THE CYLINDER AND DIFFRACTED FROM PLATE
BTEN	X,Y COMPONENTS OF UNIT VECTORS FOR RAYS TANGENT TO THE CYLINDER FROM DIFFRACTION POINT ON PLATE EDGE (FOR MOST NEGATIVE STARTING POINT ON CYLINDER)
BTCP	X,Y COMPONENTS OF UNIT VECTORS FOR RAYS TANGENT TO THE CYLINDER FROM DIFFRACTION POINT ON PLATE EDGE (FOR MOST POSITIVE STARTING POINT ON CYLINDER) ALSO SEE BTI
BTDC	X,Y COMPONENTS OF UNIT VECTORS FOR RAYS TANGENT TO THE CYLINDER FROM DIFFRACTION POINT ON PLATE EDGE (FOR FAVORED STARTING POINT ON CYLINDER)
BTI	X AND Y COMPONENTS OF SOURCE IMAGE VECTORS TANGENT TO THE CYLINDER
DTEN	DOT PRODUCT OF UNIT VECTORS OF RAYS TANGENT TO CYLINDER FROM DIFFRACTION POINT (FOR MOST NEG. STARTING REFL POINT ON CYLINDER)
DTCP	DOT PRODUCT OF UNIT VECTORS OF RAYS TANGENT TO CYLINDER FROM DIFFRACTION POINT (FOR MOST POS STARTING REFL POINT ON CYLINDER) (ALSO SEE DTI)
DTDC	DOT PRODUCT OF UNIT VECTORS OF RAYS TANGENT TO CYLINDER FROM DIFFRACTION POINT (FOR FAVORED STARTING POINT ON CYLINDER)
DTI	DOT PRODUCT OF SOURCE IMAGE VECTORS TANGENT TO THE CYLINDER (SINGLE REFLECTION FROM PLATE MP)
LCD	SET TRUE IF CORNER ME OF PLATE MP IS ON CYLINDER
LDC	SET TRUE IF EDGE ME OF PLATE MP IS STRONG DIFFRACTING PART OF WEDGE (FMP<0)
MEC	INDEX VARIABLE USED TO DETERMINE COMMON EDGES
MEN	INDEX VARIABLE USED TO DETERMINE COMMON EDGES
MEX	MAXIMUM NUMBER OF EDGES ON PLATE MP
PDCH	PHI COMPONENT OF RAY PROPAGATION DIRECTION AFTER REFLECTION FROM CYLINDER (RAY DIFFRACTED BY PLATE EDGE AND THEN REFLECTED BY CYLINDER)
PHRR	BRANCH CUT DISPLACEMENT ANGLE FOR DIFFRACTION POINT ALONG EDGE ME OF PLATE MP
RC	DISTANCE FROM Z AXIS TO PLATE CORNER
RE	RADIUS OF CYLINDER AT POINT DEFINED BY ELL ANGLE VC
TECH	THETA COMPONENT OF RAY PROPAGATION DIRECTION AFTER REFLECTION FROM CYLINDER (RAY DIFFRACTED BY PLATE EDGE AND THEN REFLECTED BY CYLINDER)
UCD	Z COMPONENT OF REFLECTION POINT LOCATION ON CYL. FOR RAY WHICH IS REFLECTED BY CYLINDER AND DIFFRACTED BY EDGE ME OF PLATE MP
UDC	Z COMPONENT OF STARTING POINT LOCATION ON CYLINDER (FOR RAY TRACING ALGORITHM) FOR RAY DIFFRACTED BY PLATE EDGE AND THEN REFLECTED BY CYLINDER
VC	ELLIPTIC ANGLE DEFINING LOCATION OF A CORNER (2-D)
VCD	ELL. ANGLE DEFINING REFLECTION POINT ON CYLINDER (2-D) FOR RAY WHICH IS REFLECTED BY CYLINDER AND DIFFRACTED BY EDGE ME OF PLATE MP
VDC	ELL ANGLE DEFINING STARTING POINT ON CYLINDER (FOR RAY-TRACING ALGORITHM) FOR RAY DIFFRACTED BY PLATE EDGE AND THEN REFLECTED BY CYLINDER
VI	X,Y,Z COMPONENTS OF PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER REFLECTION POINT
VTI	ELL ANGLE DEFINING DIRECTION OF THE TWO RAYS FROM IMAGE SOURCE TANGENT TO THE CYLINDER (SINGLE REFL. OF SOURCE RAY FROM PLATE MP)
XC	MODIFIED PLATE CORNER LOCATION USED IN DETERMINING CYLINDER REFL. PLATE DIFFRACTION LIMITS
XDC	X,Y,Z COMPONENTS OF STARTING DIFF. POINT LOCATION ON EDGE ME (FOR RAY TRACING ALGORITHM) FOR RAY DIFF. BY PLATE EDGE AND REFLECTED BY CYLINDER
XCS	X,Y,Z COMPONENTS OF STARTING REFLECTION POINT ON CYL.

CODE LISTING

```

1 C-----
2      SUBROUTINE GEOMPC
3 C!!!
4 C!!! THIS SUBROUTINE COMPUTES ALL THE GEOMETRY ASSOCIATED
5 C!!! WITH FIXED PARAMETERS FOR PLATE-CYLINDER INTERACTIONS
6 C!!!
7      DIMENSION XII(3),XIN(3),VI(3),DS(3),XC(3),VNC(3)
8      DIMENSION XOB(3),XDC(3),VTCP(2),BTCP(4),VTCN(2),BTCN(4)
9      LOGICAL LPLA,LCYL,LDC,LCD(14,6),LSRFC,LSURF,LDEBUG,LTEST
10     LOGICAL LIHD,LSHD,LSTD,LSTS,LCTD,LHCT,LHIT
11     COMMON/PIS/PI,TPI,DPR,RPD
12     COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
13     COMMON/GEJPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
14     2,MEP(14),MPX
15     COMMON/EDMAG/VMAG(14,6)
16     COMMON/SORINF/XS(3),VXS(3,3)
17     COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
18     COMMON/IMCINF/XIC(2,3),VXIC(3,3,2)
19     COMMON/FARP/IN,H,HAW
20     COMMON/SOURCE/FACTOR
21     COMMON/ENDFCL/BD(14,6,2)
22     COMMON/ENDSCL/DTS,VTS(2),BTS(4)
23     COMMON/BNDICL/DTI(14),VTI(14,2),BTI(14,4)
24     COMMON/BNDRCL/VCD(14,6),UCD(14,6),BCD(14,6,2)
25     COMMON/ENDDCL/VDC(14,6),UDC(2),PDCR(14,6,2),TDCR(14,6,2)
26     2,DTDC(14,6),BTDC(14,6,4),DDC(14,6,2)
27     COMMON/SRFACC/LSRFC(2)
28     COMMON/SURFAC/LSURF(14)
29     COMMON/LPLCY/LPLA,LCYL
30     COMMON/LSHDI/LSHD(14),LIHD(14,14)
31     COMMON/LSHDP/LSTS,LSTD(14)
32     COMMON/LDCBY/LDC(14,6)
33     COMMON/TEST/LDEBUG,LTEST
34     COMMON/FNANG/FNP(14,6)
35     COMMON/ERNPHW/PHWR(14,6)
36     IF(LDEBUG) WRITE(6,900)
37 900   FORMAT(/,' DEBUGGING GEOMPC SUBROUTINE')
38 C!!! 1. DETERMINATION OF EDGES ATTACHED TO CYLINDER
39     DO 3 MP=1,MPX
40     MEX=MEP(MP)
41     DO 3 ME=1,MEX
42     LCD(MP,ME)=.FALSE.
43 C!!! STEP THRU PLATES
44     DO 17 MP=1,MPX
45     MEX=MEP(MP)
46     MEC=0
47 C!!! STEP AROUND CORNERS ON PLATE MP
48     DO 14 ME=1,MEX
49     RC=X(MP,ME,1)*X(MP,ME,1)+X(MP,ME,2)*X(MP,ME,2)
50     VC=BTAN2(A*X(MP,ME,2),B*X(MP,ME,1))
51     XE=A*COS(VC)
52     YE=B*SIN(VC)
53     RE=XE*XE+YE*YE
54     IF(ABS(RC-RE).GT.0.01) GO TO 14
55     IF(X(MP,ME,3).GT.ZC(1)+XE*CTC(1).OR.
56     2X(MP,ME,3).LT.ZC(2)+XE*CTC(2)) GO TO 14
57     LCD(MP,ME)=.TRUE.
58     X(MP,ME,1)=XE
59     X(MP,ME,2)=YE
60     IF(MEC.NE.0) GO TO 13
61     MEC=ME
62     GO TO 14
63 13   MEN=MEC
64     IF(ME-MEC.GT.1) MEN=MEX
65 C!!! IF EDGE ME IS ATTACHED TO CYLINDER, SET WEDGE ANGLE INDICATOR
66 C!!! TO -1 AS FLAG

```

```

67      FNP(MP,MEN)=-1.
68 14    CONTINUE
69      IF(LDEBUG) WRITE(6,*) (LCD(MP,ME),ME=1,MEX)
70 17    CONTINUE
71 C!!!  2. DETERMINATION OF IMAGE BOUNDS ON CYLINDER
72 C!!!  STEP THRU PLATES
73      DO 62 MP=1,MPX
74      DO 60 N=1,3
75 60    XIN(N)=XI(MP,MP,N)
76 C!!!  CALCULATE TANGENT ANGLES AND UNIT VECTORS
77      CALL TANG(DTCP,VTCP,ETCP,XIN)
78      DTI(MP)=DTCP
79      VTI(MP,1)=VTCP(1)
80      VTI(MP,2)=VTCP(2)
81      DO 61 J=1,4
82 61    BTI(MP,J)=ETCP(J)
83      IF(.NOT.LDEBUG) GO TO 62
84      WRITE(6,*) DTI(MP)
85      WRITE(6,*) VTI(MP,1),VTI(MP,2)
86      WRITE(6,*) (BTI(MP,J),J=1,4)
87 62    CONTINUE
88 C!!!  3. DETERMINATION OF PERMISSABLE RANGE FOR CYLINDER
89 C!!!  REFLECTED, PLATE DIFFRACTED FIELD
90 C!!!  INITIALIZE VALUES
91      DO 90 MP=1,MPX
92      MEX=MEN(MP)
93      DO 90 ME=1,MEX
94      VCD(MP,ME)=0.
95      BCD(MP,ME,1)=0.
96 90    BCD(MP,ME,2)=0.
97 C!!!  STEP THRU PLATES
98      DO 92 MP=1,MPX
99 C!!!  IS PLATE MP TOTALLY SHADOWED FROM SOURCE?
100     IF(LSHD(MP)) GO TO 92
101     MEX=MEN(MP)
102 C!!!  STEP AROUND EDGES ON PLATE MP
103     DO 91 ME=1,MEX
104 C!!!  IF EDGE ME IS ON CYLINDER, DEFINE NEW CORNER
105 C!!!  SLIGHTLY OFF CYLINDER
106     IF(LCD(MP,ME)) GO TO 94
107     DO 93 N=1,3
108 93    XC(N)=X(MP,ME,N)
109     GO TO 97
110 94    VCR=BTAN2(X(MP,ME,2),X(MP,ME,1))
111     SNX=B*CLS(VCR)
112     SNY=A*SIN(VCR)
113     J=0
114 95    J=J+1
115     MJ=ME+1-J
116     IF(MJ.EQ.0) MJ=MEX
117     VCV=SNX*V(MP,MJ,1)+SNY*V(MP,MJ,2)
118     IF(ABS(VCV).LT.1.E-5) GO TO 95
119     SVCV=SIGN(.01,VCV)
120     DO 96 N=1,3
121 96    XC(N)=X(MP,ME,N)+SVCV*V(MP,MJ,N)
122 97    CONTINUE
123 C!!!  USE RAY TRACING TECHNIQUES TO DETERMINE REFL.
124 C!!!  POINT AND REFL. RAY DIRECTION OF SOURCE RAY REFL.
125 C!!!  FROM CYLINDER AND INCIDENT ON CORNER ME OF PLATE MP
126 C!!!  (SATISFY LAW OF REFLECTION)
127     CALL RFDIN(VR,UR,VI,XC)
128     VCD(MP,ME)=VR
129     UCD(MP,ME)=UR
130     DO 91 J=1,2
131     MJ=ME+1-J
132     IF(MJ.EQ.0) MJ=MEX

```

```

133      DO 91 N=1,3
134 C!!! TAKE DOT PRODUCT OF RAY INCIDENT ON CORNER AND
135 C!!! EDGE UNIT VECTOR TO OBTAIN DIFFRACTION LIMIT
136 91    BCD(MP,MJ,J)=BCD(MP,MJ,J)+V(MP,MJ,N)*VI(N)
137      IF(.NOT.LDEBUG) GO TO 92
138      WRITE(6,*) (VCD(MP,ME),ME=1,MEX)
139      WRITE(6,*) (UCD(MP,ME),ME=1,MEX)
140      WRITE(6,*) (BCD(MP,ME,1),BCD(MP,ME,2),ME=1,MEX)
141 92    CONTINUE
142 C!!! DETERMINATION OF BRANCH CUT DISPLACEMENT ANGLE FOR
143 C!!! REF-DIF AND DIF-REF TERMS
144 C!!! STEP THRU PLATES
145      DO 103 MP=1,MPX
146      MEX=MEP(MP)
147 C!!! STEP THRU EDGES
148      DO 101 ME=1,MEX
149      XPHW=X(MP,ME,1)+0.5*VMAG(MP,ME)*V(MP,ME,1)
150      YPHW=X(MP,ME,2)+0.5*VMAG(MP,ME)*V(MP,ME,2)
151      PHWR(MP,ME)=ETAN2(YPHW,XPHW)
152 101    CONTINUE
153      IF(.NOT.LDEBUG) GO TO 103
154      WRITE(6,*) (PHWR(MP,ME),ME=1,MEX)
155 103    CONTINUE
156 C!!! 4. DETERMINATION OF PARAMETERS FOR RAY DIFFRACTED
157 C!!! BY PLATE EDGE AND THEN REFLECTED OFF OF CYLINDER
158      DO 111 MP=1,MPX
159      MEX=MEP(MP)
160      DO 111 ME=1,MEX
161 111    LDC(MP,ME)=.FALSE.
162 C!!! STEP THRU PLATES
163      DO 114 MP=1,MPX
164      IF(LSHD(MP)) GO TO 114
165      MEX=MEP(MP)
166 C!!! STEP THRU EDGES
167      DO 113 ME=1,MEX
168      IF(FNP(MP,ME).LT.0.) GO TO 112
169      LDC(MP,ME)=.TRUE.
170 112    CONTINUE
171 113    CONTINUE
172      IF(LDEBUG) WRITE(6,*) (LDC(MP,ME),ME=1,MEX)
173 114    CONTINUE
174      UDC(1)=ZC(1)+A*CTC(1)
175      UDC(2)=ZC(2)+A*CTC(2)
176      IF(LDEBUG) WRITE(6,*) UDC(1),UDC(2)
177 C!!! STEP THRU PLATES
178      DO 118 MP=1,MPX
179      MEX=MEP(MP)
180 C!!! STEP THRU CORNERS
181      DO 118 ME=1,MEX
182      IF(.NOT.LDC(MP,ME)) GO TO 118
183      MJ=ME+1
184      IF(MJ.GT.MEX) MJ=1
185      VDCA=BTAN2(A*X(MP,ME,2),B*X(MP,ME,1))
186      VDCB=BTAN2(A*X(MP,MJ,2),B*X(MP,MJ,1))
187      VDC(MP,ME)=.5*(VDCA+VDCB)
188 C!!! CALCULATE (STARTING) REFLECTION POINT AT MOST
189 C!!! POSITIVE END OF CYLINDER
190      XOB(1)=A*COS(VDC(MP,ME))
191      XOB(2)=B*SIN(VDC(MP,ME))
192      XOB(3)=UDC(1)
193      VNX=B*COS(VDC(MP,ME))
194      VNY=A*SIN(VDC(MP,ME))
195 C!!! COMPUTE STARTING DIFFRACTION POINT CORRESPONDING
196 C!!! TO REFLECTION POINT AT MOST POS. END OF CYL.
197      CALL DPTNFW(XS,XOB,XDC,ME,MP)
198 C!!! CALCULATE CORRESPONDING REFLECTED RAY PROPAGATION

```

```

199 C!!! DIRECTION FOR ABOVE DIFFRACTION AND REFL. POINTS
200 VI(1)=XOB(1)-XDC(1)
201 VI(2)=XCB(2)-XDC(2)
202 VI(3)=XOB(3)-XDC(3)
203 CNIP=VN*VI(1)+VNY*VI(2)
204 CNIN=VN*VI(2)-VNY*VI(1)
205 PDCH(MP,ME,1)=BTAN2((CNIN*VIX-CNIP*VNY),-(CNIP*VN*CNIN*VNY))
206 CPDC=COS(PDCH(MP,ME,1))
207 SPDC=SIN(PDCH(MP,ME,1))
208 TDCR(MP,ME,1)=BTAN2(-CNIP,(VN*CPDC+VNY*SPDC)*VI(3))
209 C!!! CALCULATE BOUND ON REFLECTION ANGLE
210 DDC(MP,ME,1)=COS(TDCR(MP,ME,1))
211 C!!! REPEAT CALCULATIONS FOR MOST NEGATIVE ENDCAP
212 CALL TANG(DTCP,VTCP,BTCP,XDC)
213 XOB(3)=UDC(2)
214 CALL DPMFW(XS,XOB,XDC,ME,MP)
215 VI(1)=XOB(1)-XDC(1)
216 VI(2)=XCB(2)-XDC(2)
217 VI(3)=XCB(3)-XDC(3)
218 CNIP=VN*VI(1)+VNY*VI(2)
219 CNIN=VN*VI(2)-VNY*VI(1)
220 PDCH(MP,ME,2)=BTAN2((CNIN*VN-CNIP*VNY),-(CNIP*VN*CNIN*VNY))
221 CPDC=COS(PDCH(MP,ME,2))
222 SPDC=SIN(PDCH(MP,ME,2))
223 TDCR(MP,ME,2)=BTAN2(-CNIP,(VN*CPDC+VNY*SPDC)*VI(3))
224 DDC(MP,ME,2)=COS(TDCR(MP,ME,2))
225 CALL TANG(DTCN,VTCH,BTCH,XDC)
226 C!!! DECIDE WHICH STARTING LOCATION TO USE BETWEEN MOST
227 C!!! NEGATIVE AND MOST POS. END CAP VALUES
228 IF(DTCN.GT.DTCP) GO TO 116
229 DTDC(MP,ME)=DTCP
230 DO 115 J=1,4
231 115 BTDC(MP,ME,J)=BTCP(J)
232 GO TO 119
233 116 DTDC(MP,ME)=DTCN
234 DO 117 J=1,4
235 117 BTDC(MP,ME,J)=BTCH(J)
236 119 CONTINUE
237 IF(.NOT.LDEBUG) GO TO 118
238 WRITE(6,*) VDC(MP,ME),(PDCH(MP,ME,J),TDCR(MP,ME,J),J=1,2)
239 2,DTDC(MP,ME)
240 WRITE(6,*) (BTDC(MP,ME,J),J=1,4),(DDC(MP,ME,J),J=1,2)
241 118 CONTINUE
242 RETURN
243 END

```

IMAGE

PURPOSE

To determine location of source image for reflection of source ray off of plate MP. (double reflection image locations may be obtained by calling IMAGE twice; once for the source ray reflection from plate MP and once for the reflection of the ray from the image location off of the second plate.)

PERTINENT GEOMETRY

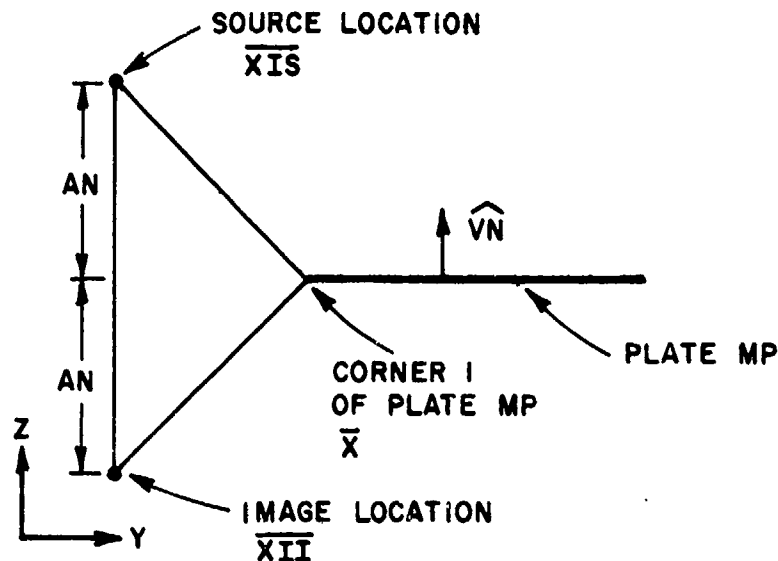


Figure 75-- Geometry for determining source image location.

$$\hat{V}N = \text{plate unit normal} = \hat{x} VN(MP,1) + \hat{y} VN(MP,2) + \hat{z} VN(MP,3)$$

$$\overline{XIS} = \hat{x} XIS(1) + \hat{y} XIS(2) + \hat{z} XIS(3)$$

$$\overline{X} = \hat{x} X(MP,1,1) + \hat{y} X(MP,1,2) + \hat{z} X(MP,1,3)$$

METHOD

The source image location is given by

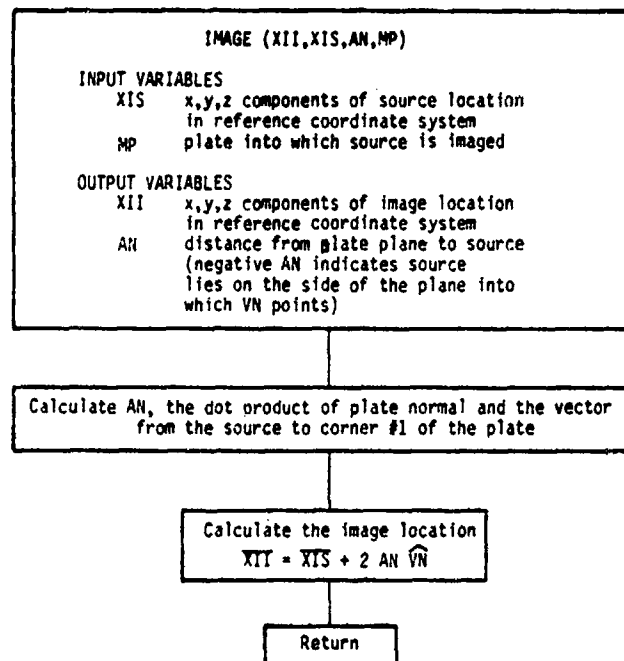
$$\overline{XII} = \overline{XIS} + 2 AN \hat{V}N = \hat{x} XII(1) + \hat{y} XII(2) + \hat{z} XII(3)$$

where

$$AN = (\overline{X} - \overline{XIS}) \cdot \hat{V}N$$

and $\overline{X}$, $\overline{XIS}$, and $\hat{V}N$ are as shown in Figure 75.

FLOW DIAGRAM



SYMBOL DICTIONARY

AN DOT PRODUCT OF VECTOR FROM SOURCE TO EDGE ONE OF
 PLATE MP AND THE PLATE UNIT NORMAL
 MP PLATE INTO WHICH SOURCE IS IMAGED
 XII X,Y,Z COMPONENTS OF IMAGE LOCATION IN RCS
 XIS X,Y,Z COMPONENTS OF SOURCE LOCATION IN RCS

CODE LISTING

```

1 C-----
2 SUBROUTINE IMAGE(XII,XIS,AN,MP)
3 C!!!
4 C!!! DETERMINE IMAGE POSITION FOR SOURCE XIS IN PLATE #MP.
5 C!!! AN INDICATES WHICH SIDE OF PLATE SOURCE IS LOCATED
6 C!!! RELATIVE TO PLATE NORMAL.
7 C!!!
8 DIMENSION XII(3),XIS(3)
9 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
10 2,MP(14),MPX
11 AN=0.
12 DO 10 N=1,3
13 10 AN=AN+(X(MP,1,N)-XIS(N))*VN(MP,N)
14 DO 20 N=1,3
15 20 XII(N)=XIS(N)+2.*AN*VN(MP,N)
16 RETURN
17 END
  
```

IMCDIR

PURPOSE

To determine the source image axes directions for a source after reflection off a given end cap.

PERTINENT GEOMETRY

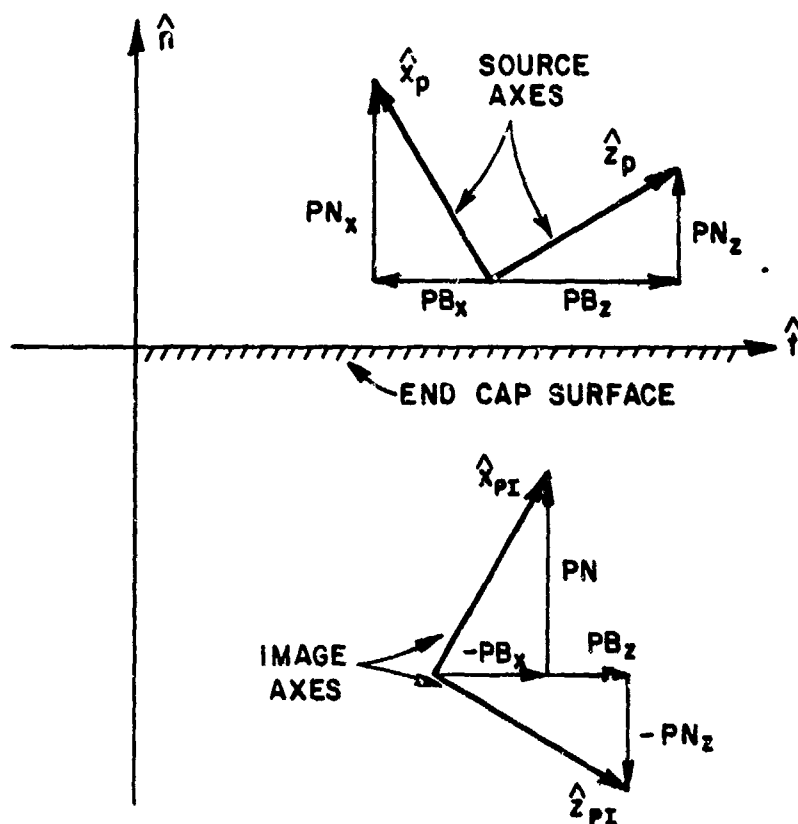


Figure 76a-- Illustration of imaging of source axes for magnetic source.

$$\hat{x}_p = \hat{x} \text{VSOURC}(1,1) + \hat{y} \text{VSOURC}(1,2) + \hat{z} \text{VSOURC}(1,3)$$

$$\hat{x}_{pi} = \hat{x} \text{VIMAG}(1,1) + \hat{y} \text{VIMAG}(1,2) + \hat{z} \text{VIMAG}(1,3)$$

$\hat{n}$ = unit normal of endcap

$\hat{t}$ = unit vector tangent to endcap

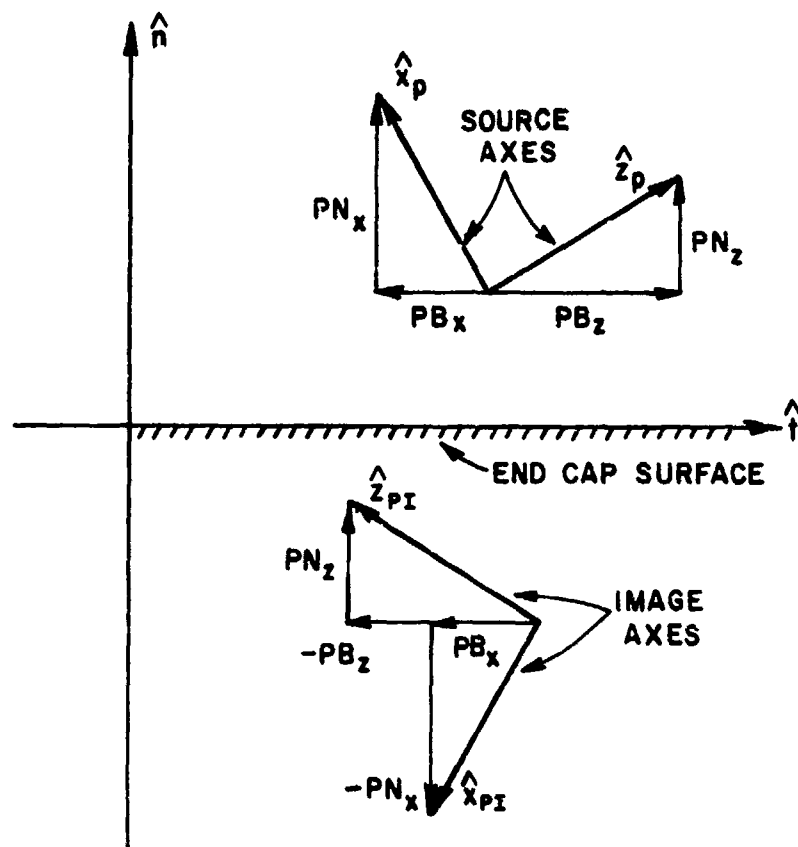


Figure 76b--Imaging of source axes for electric source.

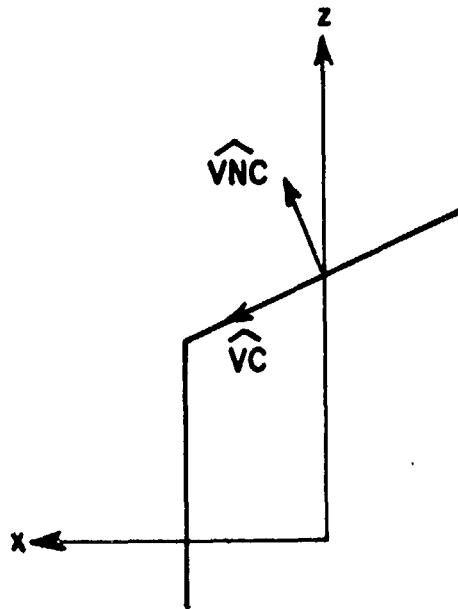


Figure 77--Endcap coordinate system.

$$\begin{aligned}\hat{n} &= \hat{VNC} = \hat{x} VNC(1) + \hat{y} VNC(2) + \hat{z} VNC(3), & VNC(2)=0 \\ \hat{t} &= \hat{VC} = \hat{x} VC(1) + \hat{y} VC(2) + \hat{z} VC(3), & VC(2)=0 \\ \hat{b} &= \hat{y}\end{aligned}$$

METHOD

The source image axes unit vectors for an electric source imaged in an end cap are given by

$$\begin{aligned}\hat{x}_{pi} &= (-\hat{x}_p \cdot \hat{n})\hat{n} + (\hat{x}_p \cdot \hat{t})\hat{t} + (\hat{x}_p \cdot \hat{b})\hat{b} \\ \hat{z}_{pi} &= (\hat{z}_p \cdot \hat{n})\hat{n} + (-\hat{z}_p \cdot \hat{t})\hat{t} + (-\hat{z}_p \cdot \hat{b})\hat{b} \\ \hat{y}_{pi} &= \hat{z}_{pi} \times \hat{x}_{pi}\end{aligned}$$

For a magnetic source, the axes are given by

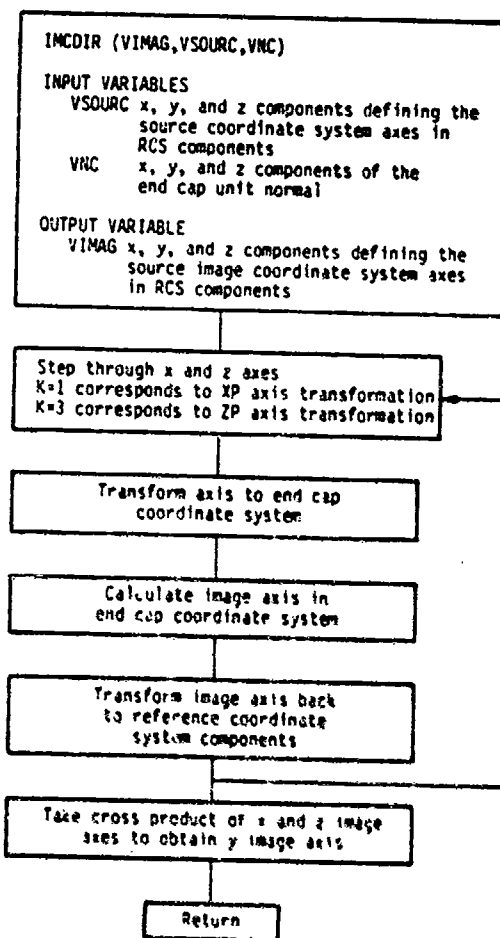
$$\hat{x}_{pi} = (\hat{x}_p \cdot \hat{n})\hat{n} + (-\hat{x}_p \cdot \hat{t})\hat{t} + (-\hat{x}_p \cdot \hat{b})\hat{b}$$

$$\hat{z}_{pi} = (-\hat{z}_p \cdot \hat{n})\hat{n} + (\hat{z}_p \cdot \hat{t})\hat{t} + (\hat{z}_p \cdot \hat{s})\hat{s}$$

$$\hat{y}_{pi} = \hat{z}_{pi} \times \hat{x}_{pi}$$

where $\hat{x}_p, \hat{y}_p, \hat{z}_p$ are unit vectors of the source coordinate system axes and $\hat{x}_{pi}, \hat{y}_{pi}, \hat{z}_{pi}$ are the unit vectors of the source image coordinate system for the end cap.

FLOW DIAGRAM



SYMBOL DICTIONARY

K INDEX VARIABLE
 L INDEX VARIABLE
 LL INDEX VARIABLE
 PB DOT PRODUCT OF END CAP UNIT BINORMAL AND UNIT VECTOR OF SOURCE AXIS BEING IMAGED
 PN DOT PRODUCT OF END CAP UNIT NORMAL AND UNIT VECTOR OF SOURCE AXIS BEING IMAGED
 PT DOT PRODUCT OF END CAP UNIT TANGENT AND UNIT VECTOR OF SOURCE AXIS BEING IMAGED
 VC X,Y,Z COMPONENTS OF UNIT VECTOR TANGENT TO END CAP (IN X-Z PLANE)
 VIMAG ARRAY OF COMPONENTS DEFINING THE SOURCE IMAGE COORDINATE SYSTEM AXES IN (X,Y,Z) REF COORD SYSTEM COMPONENTS
 VNC X,Y, AND Z COMPONENTS OF END CAP UNIT NORMAL
 VSOURC ARRAY OF COMPONENTS DEFINING THE SOURCE COORDINATE SYSTEM AXES IN (X,Y,Z) REFERENCE COORD SYS. COMPONENTS
 VX } X,Y, AND Z COMPONENTS OF SOURCE AXIS BEING IMAGED
 VY }
 VZ }

CODE LISTING

```

1 C-----
2 SUBROUTINE IMCDIR(VIMAG,VSOURC,VNC)
3 C!!!
4 C!!! DETERMINES DIRECTION OF IMAGE SOURCE COORDINATE
5 C!!! SYSTEM FOR THE END CAPS.
6 C!!!
7 DIMENSION VIMAG(3,3),VSOURC(3,3),VNC(3),VC(3)
8 COMMON/PAIP/IN,H,HAM
9 VC(1)=VNC(3)
10 VC(2)=0.
11 VC(3)=-VNC(1)
12 C!!! IMAGE X AND Z DIPOLE AXES
13 DO 15 LL=1,2
14 L=LL-1
15 K=1+2*L
16 C!!! TRANSFORM AXIS TO END CAP COORDINATE SYSTEM
17 VX=VSOURC(K,1)
18 VY=VSOURC(K,2)
19 VZ=VSOURC(K,3)
20 PR=VX*VNC(1)+VY*VNC(2)+VZ*VNC(3)
21 PT=VX*VC(1)+VY*VC(2)+VZ*VC(3)
22 PB=VY
23 C!!! FIND IMAGE AXIS
24 IF((IN+L).EQ.1) GO TO 10
25 PN=PB
26 GO TO 20
27 IO
28 PB=-PB
29 PT=-PT
30 CONTINUE
31 C!!! TRANSFORM IMAGE AXIS BACK TO REFERENCE COORDINATE SYSTEM
32 VIMAG(K,1)=PR*VNC(1)+PT*VC(1)
33 VIMAG(K,2)=PR*VNC(2)+PT*VC(2)+PB
34 VIMAG(K,3)=PR*VNC(3)+PT*VC(3)
35 CONTINUE
36 C!!! TAKE CROSS PRODUCT OF Z AND X IMAGE AXES TO
37 OBTAIN Y IMAGE AXIS
38 VIMAG(2,1)=VIMAG(3,2)*VIMAG(1,3)-VIMAG(3,3)*VIMAG(1,2)
39 VIMAG(2,2)=VIMAG(3,3)*VIMAG(1,1)-VIMAG(3,1)*VIMAG(1,3)
40 VIMAG(2,3)=VIMAG(3,1)*VIMAG(1,2)-VIMAG(3,2)*VIMAG(1,1)
41 RETURN
42 END
  
```

IMDIR

PURPOSE

To determine the image source axes directions for a source (or source image) after reflection off of a given plate.

PERTINENT GEOMETRY

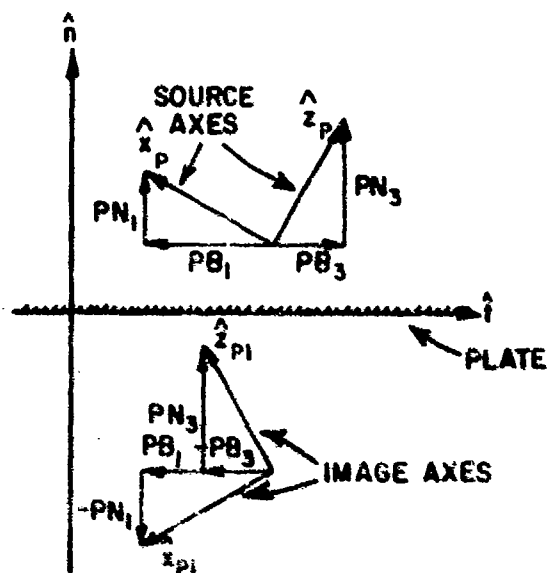


Figure 78a--Imaging of source coordinate system for electric source (shown in two dimensions for simplicity).

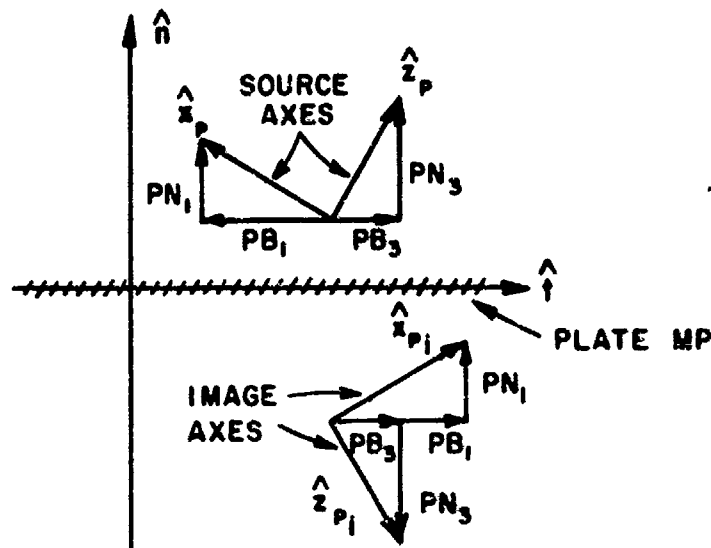


Figure 78b--Imaging of source coordinate system for magnetic source (shown in two dimensions for simplicity)

The current flows in the $\hat{z}_p$ direction.

$$\hat{n} = \text{plate unit normal} = \hat{x} V(\text{MP},1) + \hat{y} V(\text{MP},2) + \hat{z} V(\text{MP},3)$$

$$\hat{t} = \text{unit vector tangent to plate} = \hat{x} V(\text{MP},1,1) + \hat{y} V(\text{MP},1,2) + \hat{z} V(\text{MP},1,3)$$

$$\hat{b} = \hat{n} \times \hat{t} = \hat{x} VP(\text{MP},1,1) + \hat{y} VP(\text{MP},1,2) + \hat{z} VP(\text{MP},1,3)$$

(unit vectors $\hat{t}$ and $\hat{b}$ arbitrarily chosen to be edge vector $\hat{V}$ and bi-normal VP of edge #1 on the plate).

METHOD

The source image axes unit vectors for an electric source are given by

$$\hat{x}_{pi} = (-\hat{x}_p \cdot \hat{n})\hat{n} + (\hat{x}_p \cdot \hat{t})\hat{t} + (\hat{x}_p \cdot \hat{b})\hat{b}$$

$$\hat{z}_{pi} = (\hat{z}_p \cdot \hat{n})\hat{n} + (-\hat{z}_p \cdot \hat{t})\hat{t} + (-\hat{z}_p \cdot \hat{b})\hat{b}$$

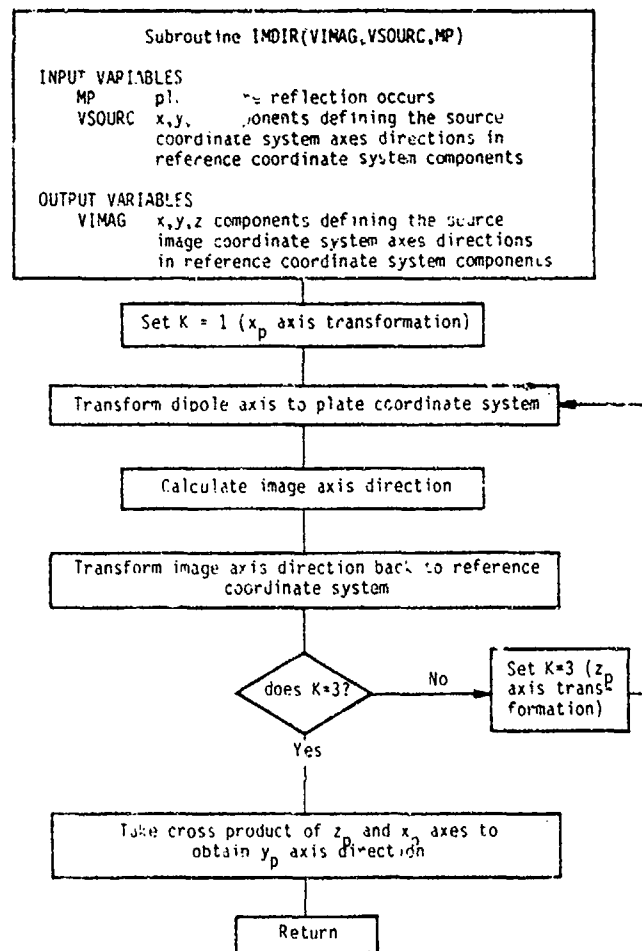
$$\hat{y}_{pi} = \hat{z}_{pi} \times \hat{x}_{pi}$$

For a magnetic source the axes are given by

$$\begin{aligned}\hat{x}_{pi} &= (\hat{x}_p \cdot \hat{n})\hat{n} + (-\hat{x}_p \cdot \hat{t})\hat{t} + (-\hat{x}_p \cdot \hat{b})\hat{b} \\ \hat{z}_{pi} &= (-\hat{z}_p \cdot \hat{n})\hat{n} + (\hat{z}_p \cdot \hat{t})\hat{t} + (\hat{z}_p \cdot \hat{b})\hat{b} \\ \hat{y}_{pi} &= \hat{z}_{pi} \times \hat{x}_{pi},\end{aligned}$$

where $\hat{x}_p, \hat{y}_p, \hat{z}_p$ are the unit vectors of the source coordinate system axes and $\hat{x}_{pi}, \hat{y}_{pi}, \hat{z}_{pi}$ are the unit vectors of the source image coordinate system.

FLOW DIAGRAM



SYMBOL DICTIONARY

K K=1 CORRESPONDS TO XP AXIS TRANSFORMATION.
 K=3 CORRESPONDS TO ZP AXIS TRANSFORMATION
 L INCREMENTAL VARIABLE
 MP PLATE OF REFLECTION
 PB COMPONENT OF AXIS IN PLATE PLANE NORMAL TO EDGE
 PN COMPONENT OF AXIS NORMAL TO PLATE
 PT COMPONENT OF AXIS PARALLEL TO PLATE EDGE
 VIMAG X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF THE
 IMAGE SOURCE COORDINATE SYSTEM AXES IN RCS
 VSOURC X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF THE
 SOURCE COORDINATE SYSTEM AXES IN RCS
 VX }
 VY } X,Y, AND Z COMPONENTS OF AXIS UNDER
 VZ } TRANSFORMATION IN RCS

CODE LISTING

```

1 C-----
2 SUBROUTINE IMDIR(VIMAG,VSOURC,MP)
3 C!!!
4 C!!! DETERMINES DIRECTION OF IMAGE SOURCE COORDINATE
5 C!!! SYSTEM FOR PLATE #MP.
6 DIMENSION VIMAG(3,3),VSOURC(3,3)
7 COMMON/FARP/IM,H,HAW
8 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
9 2,MEP(14),MPX
10 C!!! IMAGE X AND Z DIPOLE AXES
11 DO 15 LL=1,2
12 L=LL-1
13 K=1+2*L
14 C!!! TRANSFORM AXIS TO PLATE COORDINATE SYSTEM
15 VX=VSOURC(K,1)
16 VY=VSOURC(K,2)
17 VZ=VSOURC(K,3)
18 PN=VX*VN(4P,1)+VY*VN(MP,2)+VZ*VN(MP,3)
19 PT=VX*V(MP,1,1)+VY*V(MP,1,2)+VZ*V(MP,1,3)
20 PB=VX*VP(MP,1,1)+VY*VP(MP,1,2)+VZ*VP(MP,1,3)
21 C!!! FIND IMAGE AXIS
22 IF((IM+L).EQ.1) GO TO 10
23 PN=-PN
24 GO TO 20
25 10 PB=-PB
26 PT=-PT
27 20 CONTINUE
28 C!!! TRANSFORM IMAGE AXIS BACK TO REFERENCE COORDINATE SYSTEM
29 DO 15 N=1,3
30 15 VIMAG(K,N)=PN*VN(MP,N)+PT*V(MP,1,N)+PB*VP(4P,1,N)
31 C!!! TAKE CROSS PRODUCT OF Z AND X AXES TO OBTAIN Y AXIS
32 VIMAG(2,1)=VIMAG(3,2)*VIMAG(1,3)-VIMAG(3,3)*VIMAG(1,2)
33 VIMAG(2,2)=VIMAG(3,3)*VIMAG(1,1)-VIMAG(3,1)*VIMAG(1,3)
34 VIMAG(2,3)=VIMAG(3,1)*VIMAG(1,2)-VIMAG(3,2)*VIMAG(1,1)
35 RETURN
36 END

```

INCFLD

PURPOSE

To calculate the far-zone electric field transmitted by the source in a given direction with phase referred to the reference coordinate system origin.

PERTINENT GEOMETRY

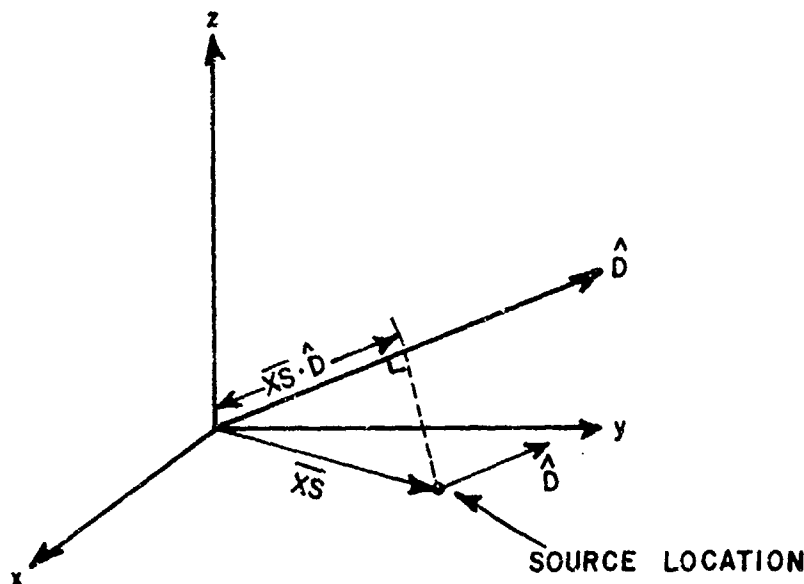


Figure 79--Geometry for source radiated field.

$$\overline{XS} = \text{source location} = \hat{x} XS(1) + \hat{y} XS(2) + \hat{z} XS(3)$$

$$\hat{D} = \text{propagation direction unit vector} = \hat{x} D(1) + \hat{y} D(2) + \hat{z} D(3)$$

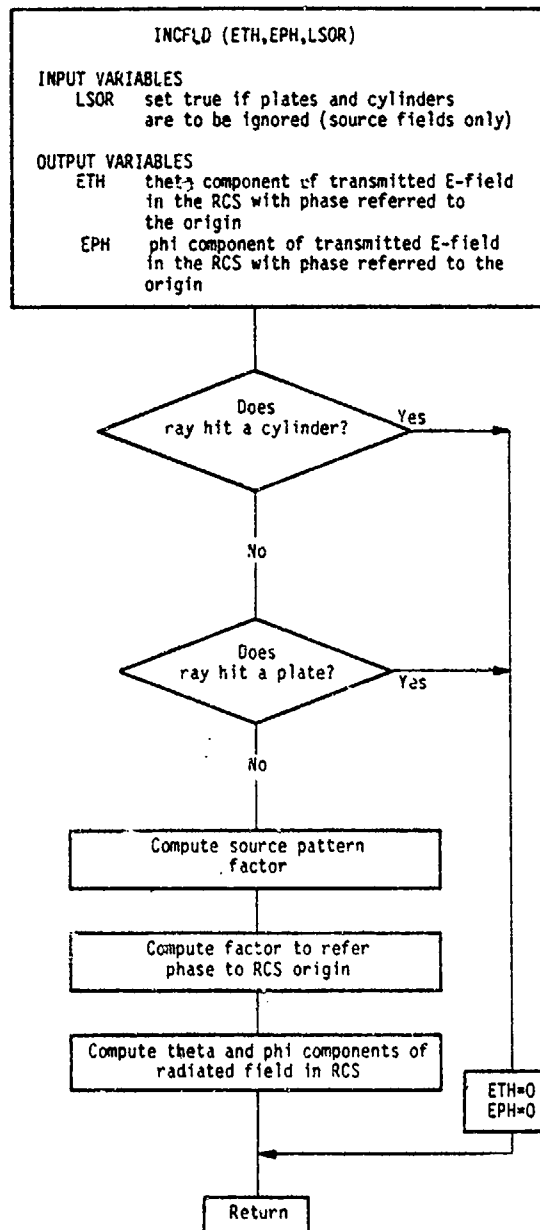
METHOD

The direct field from the source incident upon the far zone observation point is found by adding the far field phase factor $e^{jk\hat{D}\cdot\overline{XS}}$ to the source pattern factor. The existence of the field is first tested by checking if the ray from the source to the observer is shadowed by a plate or cylinder. If it is shadowed the field is set to zero. If it is not shadowed the field is given by

$$E^i(r, \theta, \phi) = W_m (ETH\hat{\theta} + EPH\hat{\phi}) \frac{e^{-jkR}}{R}$$

The factor $\frac{e^{-jkR}}{R}$ and source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM



SYMBOL DICTIONARY

D X,Y,Z COMPONENTS OF RAY PROPAGATION DIRECTION
 IN RCS
 EPH E-PHI COMPONENT OF SOURCE FIELD
 ETH E-THETA COMPONENT OF SOURCE FIELD
 LHIT SET TRUE IF RAY HITS PLATE OR CYLINDER
 PH COMPLEX PHASE CONSTANT (USED TO REFER PHASE TO
 RCS ORIGIN)
 PHSR PHI COMPONENT OF PROPAGATION DIRECTION IN RCS
 THSR THETA COMPONENT OF PROPAGATION DIRECTION IN RCS

CODE LISTING

```

1 C-----
2 SUBROUTINE INCFLD(ETH,EPH,LSOR)
3 C!!!
4 C!!! COMPUTES THE DIRECT FIELD FROM THE SOURCE WITH PHASE
5 C!!! REFERRED TO THE ORIGIN.
6 C!!!
7 COMPLEX ETH,EPH,PH,CJ,CPI4,EX,EY,EZ
8 LOGICAL LSOR,LHIT
9 COMMON/SORINF/XS(3),VXS(3,3)
10 COMMON/PIS/PI,TPI,DPR,RPD
11 COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
12 COMMON/COMP/CJ,CPI4
13 COMMON/THPHUV/DT(3),DP(2)
14 IF(LSOR)GO TO 1
15 C!!! DOES RAY HIT A CYLINDER?
16 CALL CYLINT(XS,D,PHSR,DHIT,LHIT,.FALSE.)
17 IF(LHIT) GO TO 12
18 C!!! DOES RAY HIT A PLATE?
19 CALL PLAIN(XS,D,DHIT,0,LHIT)
20 IF(LHIT) GO TO 12
21 C!!! IF RAY DOES NOT HIT ANYTHING, COMPUTE SOURCE FIELD
22 C!!! PATTERN FACTOR
23 1 CALL SOURCE(ETH,EPH,EX,EY,EZ,THSR,PHSR,VXS)
24 C!!! COMPUTE PHASE FACTOR
25 PH=CEXP(CJ*TPI*(XS(1)*D(1)+XS(2)*D(2)+XS(3)*D(3)))
26 C!!! COMPUTE THETA AND PHI COMPONENTS OF RADIATED
27 C!!! FIELD IN RCS
28 ETH=PH*ETH
29 EPH=PH*EPH
30 RETURN
31 12 ETH=(0.,0.)
32 EPH=(0.,0.)
33 RETURN
34 END
  
```

NANDB

PURPOSE

To calculate the unit vectors for rays normal and tangent to the elliptical cylinder at a given point $\overline{XC}$ (in x-y plane) defined by elliptic angle VR .

PERTINENT GEOMETRY

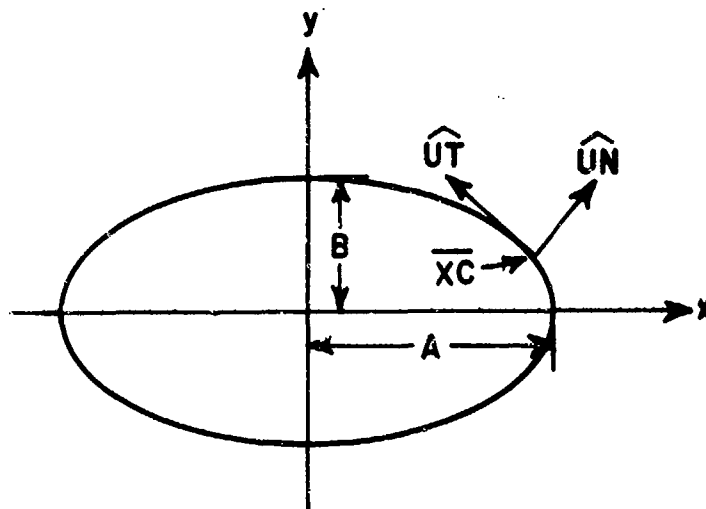


Figure 80--Illustration of unit vectors tangent and normal to the cylinder.

$$\hat{UT} = \hat{x} UT(1) + \hat{y} UT(2)$$

$$\hat{UN} = \hat{x} UN(1) + \hat{y} UN(2)$$

$$\overline{XC} = \hat{x} A \cos(VR) + \hat{y} B \sin(VR)$$

METHOD

For the point on the cylinder defined by the elliptic angle VR, the unit normal vector is given as

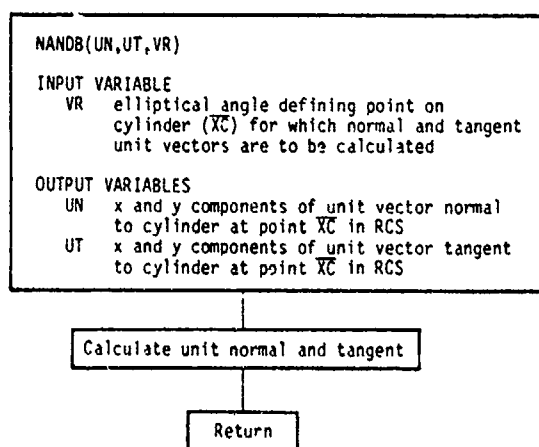
$$\hat{UN} = \frac{\hat{x} B \cos(VR) + \hat{y} A \sin(VR)}{\sqrt{B^2 \cos^2(VR) + A^2 \sin^2(VR)}}$$

and the unit tangent vector is given by

$$\hat{UT} = \frac{-\hat{x} A \sin(VR) + \hat{y} B \cos(VR)}{\sqrt{B^2 \cos^2(VR) + A^2 \sin^2(VR)}}$$

as shown in Figure 80.

FLOW DIAGRAM



SYMBOL DICTIONARY

UN	X AND Y COMPONENTS OF UNIT VECTOR NORMAL TO CYLINDER IN RCS
UT	X AND Y COMPONENTS OF UNIT VECTOR TANGENT TO CYLINDER IN RCS
VR	ELL ANGLE IN ERCS DEFINING THE POINT ON CYLINDER FOR WHICH NORMAL AND TANGENT UNIT VECTORS ARE TO BE CALCULATED

CODE LISTING

```

1 C-----
2 SUBROUTINE NADB(UN,UT,VR)
3 C!!!
4 C!!! COMPUTES NORMAL AND TANGENT VECTOR AT ANGLE VR ON THE
5 C!!! ELLIPTIC CYLINDER.
6 C!!!
7 DIMENSION UN(2),UT(2)
8 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
9 DN=SQRT(A*A*SIN(VR)*SIN(VR)+B*B*COS(VR)*COS(VR))
10 UN(1)=B*COS(VR)/DN
11 UN(2)=A*SIN(VR)/DN
12 UT(1)=-UN(2)
13 UT(2)=UN(1)
14 RETURN
15 END

```

OUTPUT

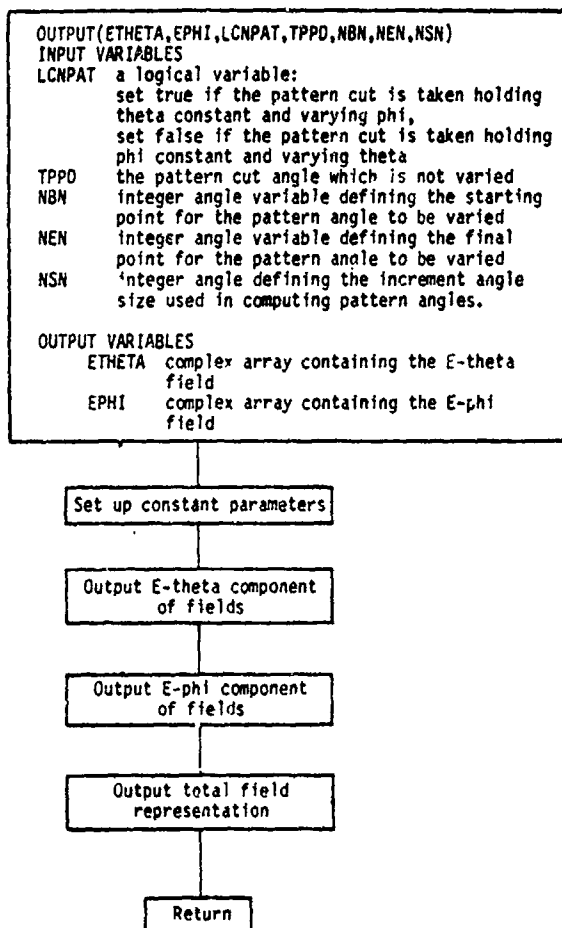
PURPOSE

To output various representations of the computed fields on the line printer.

METHOD

This subroutine outputs various representations of the fields on the line printer for a convenient analysis of the data calculated for a given pattern computation. The fields are represented in complex form, magnitude and phase, normalized and unnormalized, and in decibels. If the far field range is specified the fields are output in volts/meter. If no range is specified the fields are given in volts/unit. If the power radiated is specified the directive gain is given. If it is not specified the radiation intensity is output instead. Also, the major and minor components of the total fields are given, as well as the axial ratio and tilt angle of the polarization ellipse. Complete details of the output presentation are given in the User's Manual[8].

FLOW DIAGRAM



SYMBOL DICTIONARY

AXRAT AXIAL RATIO OF POLARIZATION ELLIPSE
 EDIF2 COMPUTATIONAL VARIABLE
 EMAJ2 MAJOR AXIS RADIATION INTENSITY *2*Z0
 EMIN2 MINOR AXIS RADIATION INTENSITY *2*Z0
 EPHA PHASE OF EPHI
 EPHDB E-PHI DIRECTIVE GAIN OR RADIATION INTENSITY
 EPHDBN NORMALIZED E-PHI GAIN OR INTENSITY
 EPHI COMPLEX ARRAY CONTAINING THE E-PHI FIELD
 EPHM MAGNITUDE OF EPHI
 EPHAN NORMALIZED E-PHI MAGNITUDE
 EPHMR MAGNITUDE OF EPHM WITH RANGE FACTOR
 EPHMX MAXIMUM MAGNITUDE OF EPHI
 EPHPS PHASE OF EPHR
 EPHR EPHI WITH RANGE FACTOR INCLUDED
 ETD1M MAXIMUM MAGNITUDE OF THE RADIATION INTENSITY*2*Z0
 ETHA PHASE OF ETHETA
 ETHDB E-THETA DIRECTIVE GAIN OR RADIATION INTENSITY
 ETHDBN THEIA NORMALIZED GAIN OR INTENSITY
 ETHETA COMPLEX ARRAY CONTAINING THE E-THETA FIELD
 ETHM MAGNITUDE OF ETHETA
 ETHMN NORMALIZED E-THETA MAGNITUDE
 ETHMR MAGNITUDE OF ETHM WITH RANGE FACTOR
 ETHMX MAXIMUM MAGNITUDE OF ETHETA
 ETHPS PHASE OF ETHR
 ETHR ETHETA WITH RANGE FACTOR
 ETC12 RADIATION INTENSITY TIMES 2*Z0
 ETC1N NORMALIZED GAIN OR INTENSITY
 FACP GAIN OR INTENSITY FACTOR
 FACPDB FACP IN DB
 FRANG RANGE FACTOR
 GLURBA COMPUTATIONAL VARIABLE
 GMAJ MAJOR AXIS DIRECTIVE GAIN OR RADIATION INTENSITY IN DB
 GMIN MINOR AXIS DIRECTIVE GAIN OR RADIATION INTENSITY IN DB
 GTC1 DIRECTIVE GAIN OR RADIATION INTENSITY IN DB
 GTC1N NORMALIZED GAIN OR INTENSITY IN DECIBELS
 I DO LCOP INDEX
 IM INTEGER VALUE OF ANGLE BEING VARIED
 IMAX NUMBER OF LINES TO BE OUTPUT BETWEEN SPACING
 LCNPA1 LOGICAL VARIABLE RELATED TO THE PATTERN CUT TAKEN:
 LCNPAT=TRUE IF THETA IS FIXED AND PHI IS VARIED, AND
 LCNPAT=FALSE IF PHI IS FIXED AND THETA IS VARIED
 NDA ONE PLUS NBN
 NBN AN INTEGER DEFINING THE STARTING POINT OF PATTERN
 ANGLE TO BE VARIED
 NEM ONE PLUS NEN
 NEN AN INTEGER DEFINING THE ENDING POINT OF THE PATTERN
 ANGLE WHICH IS VARIED
 NSI AN INTEGER DEFINING THE INCREMENT IN THE PATTERN
 ANGLE WHICH IS VARIED BETWEEN STARTING AND END POINTS
 PHI FIXED PHI ANGLE
 RANCL RANGE PHASE VALUE
 STILT1 SINE OF TILTA
 THI FIXED THETA ANGLE
 TILTA TILT ANGLE OF POLARIZATION ELLIPSE IN RADIAN
 TILT1 TILT ANGLE IN DEGREES
 TPDF THE FIXED ANGLE DEFINING THE PATTERN CUT

1
1
1
1
1

```

47 152 FORMAT (6X,'THETA',9X,'PHI',16X,'E-PHI',14X,'PHASE',7X,
48 2'MAGNITUDE',4X,'DB GAIN',6X,'MAGNITUDE',7X,'DB')
49 153 FORMAT (6X,'THETA',9X,'PHI',15X,'E-THETA',13X,'PHASE',7X,
50 2'MAGNITUDE',4X,'DB GAIN',6X,'MAGNITUDE',7X,'DB')
51 155 FORMAT (6X,'THETA',9X,'PHI',15X,'E-THETA',13X,'PHASE',7X,
52 2'MAGNITUDE',3X,'DB INTEN.',5X,'MAGNITUDE',7X,'DB')
53 156 FORMAT (6X,'THETA',9X,'PHI',16X,'E-PHI',14X,'PHASE',7X,
54 2'MAGNITUDE',3X,'DB INTEN.',5X,'MAGNITUDE',7X,'DB')
55 154 FORMAT (2(3X,'-----'),3X,'-----',
56 2,'-----',3X,'-----',2(3X,'-----',
57 2'-----')/)
58 200 FORMAT (1H1)
59 400 FORMAT (1H0)
60 110 FORMAT (3X,9(F10.5,3X))
61 201 FORMAT(2(3X,F10.5),2(2X,E11.5),3X,F10.5,2X,E11.5,3(3X,F10.5))
62 C!!! SET UP CONSTANTS
63 NBM=NBH+1
64 NEM=NEH+1
65 IF(LCNPAT) TH1=TPPD
66 FRANG=CMPLX(1.,0.)
67 IF(.NOT.LRANG) GO TO 600
68 RANGL=RANG/VL-AINT(RANG/VL)
69 FRANG=CEXP(CMPLX(0.,-TPI*RANGL))/RANG
70 600 CONTINUE
71 FACP=1./(240.*PI)
72 IF(LPRAD) FACP=1./(60.*PRAD)
73 FACPDB=10.*BLOG10(FACP)
74 ETHMX = EABS(ETHETA(1))
75 EPHMX = EABS(EPHI(1))
76 ETOTMX=ETHMX*ETHMX+EPHMX*EPHMX
77 DO 1 I = NBM,NEM,NSH
78 ETHI = EABS(ETHETA(I))
79 IF (ETHI.GT.ETHMX) ETHMX = ETHI
80 EPHI = EABS(EPHI(I))
81 IF (EPHI.GT.EPHMX) EPHMX = EPHI
82 ETOT2=ETHI*ETHI+EPHI*EPHI
83 IF(ETOT2.GT.ETOTMX) ETOTMX=ETOT2
84 CONTINUE
85 C!!! OUTPUT E-THETA REPRESENTATIONS
86 WRITE(6,300)
87 WRITE(6,100)
88 WRITE (6,100)
89 WRITE (6,101)
90 WRITE (6,102)
91 WRITE (6,103)
92 WRITE (6,104)
93 WRITE (6,105)
94 WRITE (6,106)
95 WRITE (6,107)
96 WRITE (6,150)
97 WRITE(6,151)
98 IF(LPRAD) WRITE (6,153)
99 IF(.NOT.LPRAD) WRITE(6,155)
100 WRITE (6,154)
101 IMAX=10*NSH+1
102 DO 2 I = NBM,NEM,NSH
103 IM=I-1
104 IF(LCNPAT) PHI=IM
105 IF(LCNPAT) GO TO 25
106 IF(1.GT.180) GO TO 24
107 PHI=TPPD
108 PHI=IM
109 GO TO 25
110 PHI=TPPD+180.
111 IF(PHI.GE.360.) PHI=PHI-360.
112 PHI=360-PHI

```

```

112 25 CONTINUE
114 ETHR=ETHETA(I)*FRANG
115 ETHM = LABS(ETHETA(I))
116 ETHMR=ETHM/RANG
117 ETHPS = DPR*BTAN2(AIMAG(ETHR),REAL(ETHR))
118 ETHDB = 20.*BLOG10(ETHM)+FACPD
119 ETHMN = ETHM/ETHMX
120 ETHDBN = 20.*BLOG10(ETHMN)
121 IF (I.GT. IMAX) IMAX = IMAX+10*NSN
122 WRITE (6,501) THI,PHI,ETHR,ETHPS,ETHMR,ETHDB,ETHMN,ETHDBN
123 IF (I.LE. IMAX) WRITE (6,400)
124 2 CONTINUE
125 C!!! OUTPUT E-PHI REPRESENTATIONS
126 WRITE(6,100)
127 WRITE (6,120)
128 WRITE (6,300)
129 WRITE(6,100)
130 WRITE (6,100)
131 WRITE (6,201)
132 WRITE (6,202)
133 WRITE (6,203)
134 WRITE (6,204)
135 WRITE (6,205)
136 WRITE (6,206)
137 WRITE (6,207)
138 WRITE (6,150)
139 WRITE (6,151)
140 IF(LPRAD) WRITE (6,152)
141 IF(.NOT.LPRAD) WRITE(6,156)
142 WRITE (6,154)
143 IMAX=10*NSN+1
144 DO 3 I = NEM,NEM,NSN
145 IM=I-1
146 IF(LCNPAT) PHI=IM
147 IF(LCNPAT) GO TO 35
148 IF(IM.GT.180) GO TO 34
149 PHI=TPPD
150 THI=IM
151 GO TO 35
152 34 PHI=TPPD+180.
153 IF(PHI.GE.360.) PHI=PHI-360.
154 THI=360-IM
155 35 CONTINUE
156 EPHI=EPHI(I)*FRANG
157 EPHM = LABS(EPHI(I))
158 EPHMR=EPHM/RANG
159 EPHPS = DPR*BTAN2(AIMAG(EPHR),REAL(EPHR))
160 EPHDB = 20.*BLOG10(EPHM)+FACPD
161 EPHMN = EPHM/EPHMX
162 EPHDBN = 20.*BLOG10(EPHMN)
163 IF (I.GT. IMAX) IMAX = IMAX+10*NSN
164 WRITE (6,501) THI,PHI,EPHR,EPHPS,EPHMR,EPHDB,EPHMN,EPHDBN
165 IF (I.LE. IMAX) WRITE (6,400)
166 2 CONTINUE
167 C!!! OUTPUT TOTAL FIELD REPRESENTATIONS
168 WRITE (6,100)
169 WRITE(6,100)
170 WRITE(6,300)
171 WRITE(6,100)
172 WRITE(6,100)
173 IF(LPRAD) WRITE(6,301)
174 301 FORMAT(' TOTAL DIRECTIVE GAIN IN DB.////')
175 IF(.NOT.LPRAD) WRITE(6,303)
176 303 FORMAT(' TOTAL RADIATION INTENSITY IN DB.////')
177 WRITE(6,150)
178 IF(LPRAD) WRITE(6,302)

```

```

179 362 FORMAT(6X,'THETA',9X,'PHI',9X,'MAJOR',8X,'MINOR',7X
180 2,'TILT ANG',4X,'AXIAL RATIO',2X,'TOTAL GAIN',4X,'NORM GAIN')
181 IF(.NOT.LPRAD) WRITE(6,364)
182 364 FORMAT(6X,'THETA',9X,'PHI',9X,'MAJOR',8X,'MINOR',7X
183 2,'TILT ANG',4X,'AXIAL RATIO',2X,'TOTAL INTEN.',2X,'NORM',
184 2,'INTEN.')
185 IMAX=10*NSN+1
186 DO 4 I=NSM,NEM,NSN
187 IM=I-1
188 IF(LCHPAT) PHI=IM
189 IF(LCHPAT) GO TO 45
190 IF(IM.GT.180) GO TO 44
191 PHI=TPPD
192 THI=IM
193 GO TO 45
194 44 PHI=TPPD+180.
195 IF(PHI.GE.360.) PHI=PHI-360.
196 THI=360-IM
197 45 CONTINUE
198 ETHM=BAES(ETHETA(I))
199 EPHM=BAES(EPhi(I))
200 ETOT2=ETHM*ETHM+EPHM*EPHM
201 GTOT=10.*BLOG10(FACP*ETOT2)
202 ETOTN=ETOT2/ETOTM
203 GTOTN=10.*BLOG10(ETOTN)
204 IF(1.GT.IMAX) IMAX=IMAX+10*NSN
205 EPHA=BTAN2(AIMAG(EPhi(I)),REAL(EPhi(I)))
206 ETHA=BTAN2(AIMAG(ETHETA(I)),REAL(ETHETA(I)))
207 GLURBA=2.*EPHM*ETHM*COS(EPHA-ETHA)
208 EDIF2=ETHM*ETHM-EPHM*EPHM
209 TILTA=.5*BTAN2(GLURBA,EDIF2)
210 TILTD=DPR*TILTA
211 STILTA=SIN(TILTA)
212 EIAJ2=-EDIF2*STILTA*STILTA+GLURBA*STILTA*COS(TILTA)+
213 2ETHM*ETHM
214 GMAJ=10.*BLOG10(EIAJ2)+FACPDB
215 EMIN2=EDIF2*STILTA*STILTA-GLURBA*STILTA*COS(TILTA)+
216 2EPHM*EPHM
217 GMIN=10.*BLOG10(EMIN2)+FACPDB
218 AXRAT=SQRT(ABS(EMIN2/EMAJ2))
219 WRITE(6,500) THI,PHI,GMAJ,GMIN,TILTD,AXRAT,GTOT,GTOTN
220 IF(1.EQ.IMAX) WRITE(6,400)
221 4 CONTINUE
222 WRITE(6,100)
223 WRITE(6,100)
224 RETURN
225 END

```

PATROT

PURPOSE

To convert pattern angles from pattern cut coordinate system to reference coordinate system representation.

PERTINENT GEOMETRY

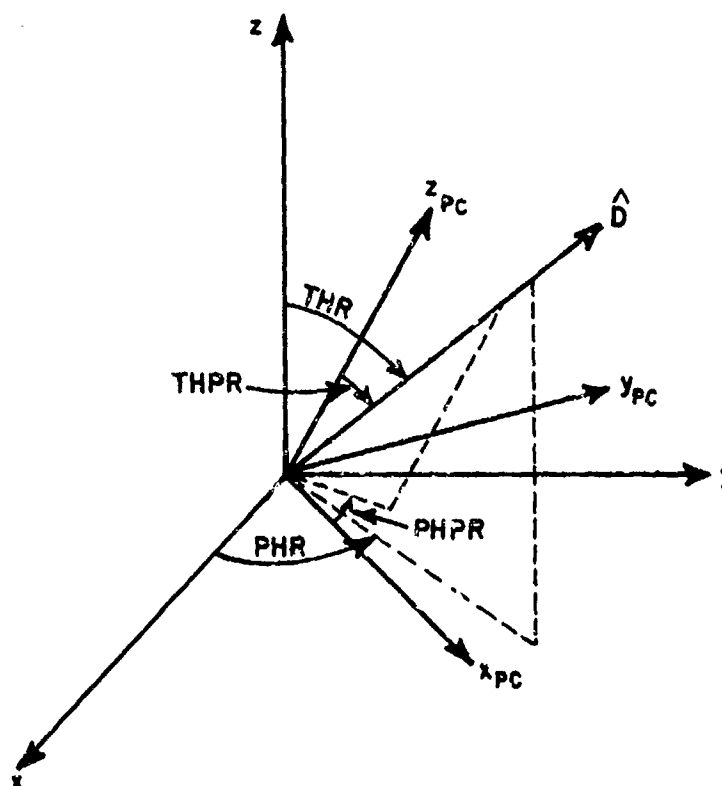


Figure 81--Illustration of propagation direction $\hat{D}$ and reference and pattern-cut coordinate systems.

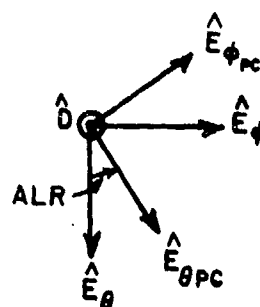


Figure 82--Illustration of polarization rotation angle ALR .

METHOD

The observation direction is defined in the pattern cut coordinate system as

$$\hat{D} = \cos(\text{PHPR})\sin(\text{THPR})\hat{x}_p + \sin(\text{PHPR})\sin(\text{THPR})\hat{y}_p + \cos(\text{THPR})\hat{z}_p.$$

This is converted into the reference coordinate system as

$$\hat{D} = (\hat{D} \cdot \hat{x})\hat{x} + (\hat{D} \cdot \hat{y})\hat{y} + (\hat{D} \cdot \hat{z})\hat{z}$$

or

$$\hat{D} = \cos(\text{PHR})\sin(\text{THR})\hat{x} + \sin(\text{PHR})\sin(\text{THR})\hat{y} + \cos(\text{THR})\hat{z}.$$

The polarization conversion angle is given by

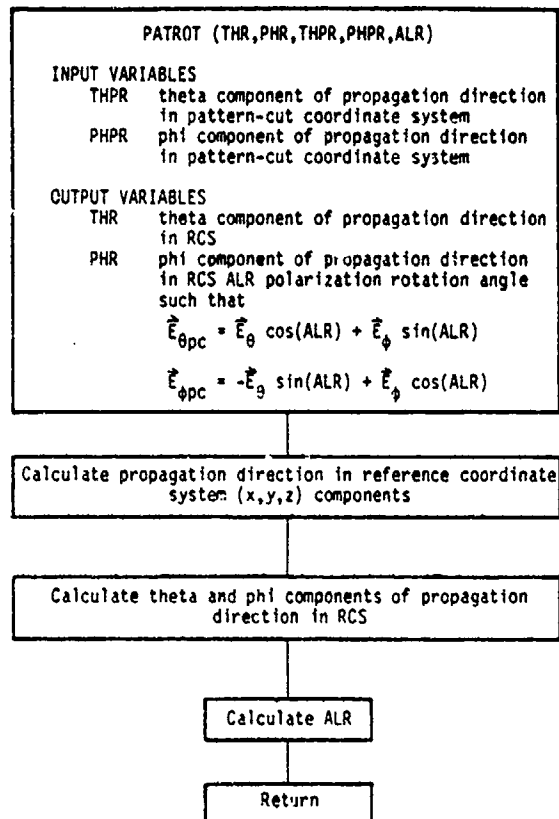
$$\text{ALR} = \tan^{-1} \frac{\hat{\theta}_{pr} \cdot \hat{\phi}}{\hat{\theta}_{pc} \cdot \hat{\theta}}$$

so that after the scattered fields are computed they can be converted back to the pattern cut coordinate system using

$$\vec{E}_{\theta pc} = \vec{E}_{\theta} \cos(\text{ALR}) + \vec{E}_{\phi} \sin(\text{ALR})$$

$$\vec{E}_{\phi pc} = -\vec{E}_{\theta} \sin(\text{ALR}) + \vec{E}_{\phi} \cos(\text{ALR}) .$$

FLOW DIAGRAM



SYMBOL DICTIONARY

ALR	POLARIZATION ROTATION ANGLE
CPH	COS(PHR)
CPHP	COS(PHPR)
CTH	COS(THR)
CTHP	COS(THPR)
PDTP	COMPUTATIONAL VARIABLE
PHPR	PHI COMPONENT OF PROPAGATION DIRECTION IN PATTERN CUT COORDINATE SYSTEM
PHR	PHI COMPONENT OF PROPAGATION DIRECTION IN RCS
RDX	X, Y, AND Z COMPONENTS OF PROPAGATION DIRECTION IN RCS
RDY	
RDZ	
SPH	SIN(PHR)
SPHP	SIN(PHPR)
STH	SIN(THR)
STHP	SIN(THPR)
TDTP	COMPUTATIONAL VARIABLE
THPR	THETA COMPONENT OF PROPAGATION DIRECTION IN PATTERN CUT COORD SYSTEM
THR	THETA COMPONENT OF PROPAGATION DIRECTION IN RCS
TX	X, Y, Z COMPONENTS OF THETA POLARIZATION UNIT VECTOR OF PATTERN CUT COORDINATE SYSTEM IN RCS COMPONENTS
TY	
TZ	

CODE LISTING

```

1 C-----
2      SUBROUTINE PATROT(THR,PHR,THPR,PHPR,ALR)
3 C!!!
4 C!!!
5 C!!! ROTATION OF PATTERN ANGLES FROM PATTERN AXES (THP,PHP)
6 C!!! TO REFERENCE AXES (TH,PH). NOTE THAT ALR IS DEFINED BY:
7 C!!!   E-THETAP=E-THETA*COS(ALR)+E-PHI*SIN(ALR)
8 C!!!   E-PHIP=-E-THETA*SIN(ALR)+E-PHI*COS(ALR)
9 C!!!
10 C!!!
11      LOGICAL LDEBUG,LTEST
12      COMMON/TEST/LDEBUG,LTEST
13      COMMON/PI/PI,TPI,DPR,RPD
14      COMMON/PATDAT/XPC(3),YPC(3),ZPC(3)
15      STHP=SIN(THPR)
16      CTHP=COS(THPR)
17      SPHP=SIN(PHPR)
18      CPHP=COS(PHPR)
19 C!!! CALCULATE PROPAGATION DIRECTION IN REFERENCE COORDINATE
20 C!!! SYSTEM (X,Y,Z) COORDINATES
21      RDX=STHP*CPHP*XPC(1)+STHP*SPHP*YPC(1)+CTHP*ZPC(1)
22      RDY=STHP*CPHP*XPC(2)+STHP*SPHP*YPC(2)+CTHP*ZPC(2)
23      RDZ=STHP*CPHP*XPC(3)+STHP*SPHP*YPC(3)+CTHP*ZPC(3)
24      SON=SQRT(RDX*RDX+RDY*RDY)
25 C!!! CALCULATE THR AND PHR
26      THR=ATAN2(SON,RDX)
27      PHR=ATAN2(RDY,RDX)
28      STH=SIN(THR)
29      CTH=COS(THR)
30      SPH=SIN(PHR)
31      CPH=COS(PHR)
32      TX=CTHP*CPHP*XPC(1)+CTHP*SPHP*YPC(1)-STHP*ZPC(1)
33      TY=CTHP*CPHP*XPC(2)+CTHP*SPHP*YPC(2)-STHP*ZPC(2)
34      TZ=CTHP*CPHP*XPC(3)+CTHP*SPHP*YPC(3)-STHP*ZPC(3)
35 C!!! CALCULATE ALR
36      TDTP=TX*CTH*CPH+TY*CTH*SPH-TZ*STH
37      PDTP=-TX*SPH+TY*CPH
38      ALR=ATAN2(PDTP,TDTP)
39      IF (.NOT.LTEST) GO TO 1
40      WRITE(6,2)
41 2      FORMAT(/,' TESTING PATROT SUBROUTINE')
42      WRITE(6,*) THR,PHR,THPR,PHPR,ALR
43 1      RETURN
44      END

```

PFUN

PURPOSE

This function computes the p^* function for the cylinder's acoustically soft diffraction coefficient.

METHOD

The p^* function is defined as [14,15]

$$p^*(x) = \frac{1}{2\sqrt{\pi x}} + \hat{p}_s(x) e^{j\pi/4}$$

where

$$\hat{p}_s(x) = \frac{e^{-j\pi/4}}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{V(\tau)}{w_2(\tau)} e^{-jx\tau} d\tau,$$

and $V(\tau)$ and $w_2(\tau)$ are Fock type Airy functions. The p^* function is computed as follows:

1) for $x \leq -3$

$$p^*(x) = \frac{1}{2\sqrt{\pi x}} + \frac{1}{2}\sqrt{|x|} \left(1 + j\frac{2}{x}\right) e^{j\frac{x^3}{12}} e^{j\pi/4}$$

2) for $-3 < x < 2$

$$p^*(x) = p^*(x_i) + \frac{(x-x_i)}{(x_{i+1}-x_i)} (p^*(x_{i+1}) - p^*(x_i)),$$

where the $p^*(x_i)$ are tabulated values [14,15] and $x_{i+1} - x_i = 0.1$ with $x_i \leq x \leq x_{i+1}$.

3) For $x \geq 2$

$$p^*(x) = \frac{1}{2\sqrt{\pi x}} - \frac{e^{j\pi/6}}{2\sqrt{\pi}} \sum_{n=1}^5 \frac{e^{xq_n} e^{-j\frac{5\pi}{6}}}{[A'_1(-q_n)]^2}$$

where $A'_1(\tau)$ is the derivative of the Miller type Airy function.

SYMBOL DICTIONARY

AMC $-0.5 * \text{CEXP}(J * \pi / 6) / \text{SQRT}(\pi)$
 AQ DERIVATIVE OF MILLER TYPE AIRY FUNCTION AT Q
 C $0.5 / \text{SQRT}(\pi)$
 EXC $\text{CEXP}(-5 * \pi / 6)$
 I SMALLEST INTEGER CLOSEST TO $10 * X$
 Q ZERGES OF MILLER TYPE AIRY FUNCTION
 PFUN P FUNCTION
 PJ IMAGINARY PART OF TABULATED P FUNCTION
 PR REAL PART OF TABULATED P FUNCTION
 X ARGUMENT OF P FUNCTION
 XI REAL NUMBER REPRESENTATION OF I

CODE LISTING

```

1 C-----
2     COMPLEX FUNCTION PFUN(X)
3     DIMENSION PR(51),PJ(51)
4 C!!!
5 C!!! COMPUTES THE P FUNCTION OF THE CYLINDER'S
6 C!!! DIFFRACTION COEFFICIENT (SOFT CASE)
7 C!!!
8     COMPLEX AMC,EXC
9     DIMENSION Q(5),AQ(5)
10    COMMON/PIS/PI,TPI,DPR,RPD
11    DATA AMC,EXC/(-0.24430,-0.14105),(-0.866025,-0.5)/
12    DATA C/0.28209/
13    DATA Q/2.33811,4.08795,5.52056,6.78671,7.94413/
14    DATA AQ/0.70121,-0.80311,0.86520,-0.91085,0.94734/
15    DATA PR/-.054,.125,.276,.399,.475,.560,.605,.629,.638,.636
16    2,.624,.606,.584,.560,.536,.510,.487,.464,.444,.425,.408
17    2,.393,.379,.367,.357,.347,.338,.330,.322,.314,.307,.299
18    2,.292,.284,.276,.268,.259,.251,.242,.234,.224,.215,.206
19    2,.198,.190,.180,.173,.165,.158,.150,.144/
20    DATA PJ/.879,.840,.769,.678,.577,.469,.354,.265,.173,.091
21    2,.019,-.043,-.113,-.139,-.174,-.202,-.224,-.240,-.251
22    2,-.257,-.260,-.260,-.256,-.252,-.244,-.236,-.225,-.214
23    2,-.212,-.190,-.177,-.164,-.151,-.138,-.125,-.113,-.101
24    2,-.090,-.080,-.070,-.061,-.053,-.045,-.039,-.032,-.027
25    2,-.023,-.018,-.014,-.011,-.010/
26    IF(X.LE.-3.)GO TO 1
27    IF(X.GE.2.)GO TO 2
28    I=((3.+X)*10.)
29    XI=FLOAT(I)-30.
30    I=I+1
31    PFUN=CMPLX(PR(I),-PJ(I))+ (10.*X-XI)*CMPLX(PR(I+1)-PR(I),
32    2-PJ(I+1)+PJ(I))
33    RETURN
34 1    PFUN=.5*(1./SQRT(PI)*X)+SQRT(AFS(X))*CEXP(CMPLX(C,.625
35    2*PI+X*X*X/12.))*CMPLX(1.,2./(X*X*X))
36    RETURN
37 2    PFUN=(0.,0.)
38    DO 3 N=1,5
39    PFUN=PFUN+CEXP(X*Q(N)*EXC)/AQ(N)/AQ(N)
40    PFUN=PFUN+AMC+C/X
41    RETURN
42    END
  
```

PLAIN

PURPOSE

To determine if a ray traveling from a given source location in a given direction will intersect a given plate (or set of plates).

Note: several modes of operation are available:

If $MH = -MP$ then only plate MP is checked ($MP > 0$)
If $MH = 0$ all plates are checked
If $MH = MP$ all plates except plate MP are checked.

PERTINENT GEOMETRY

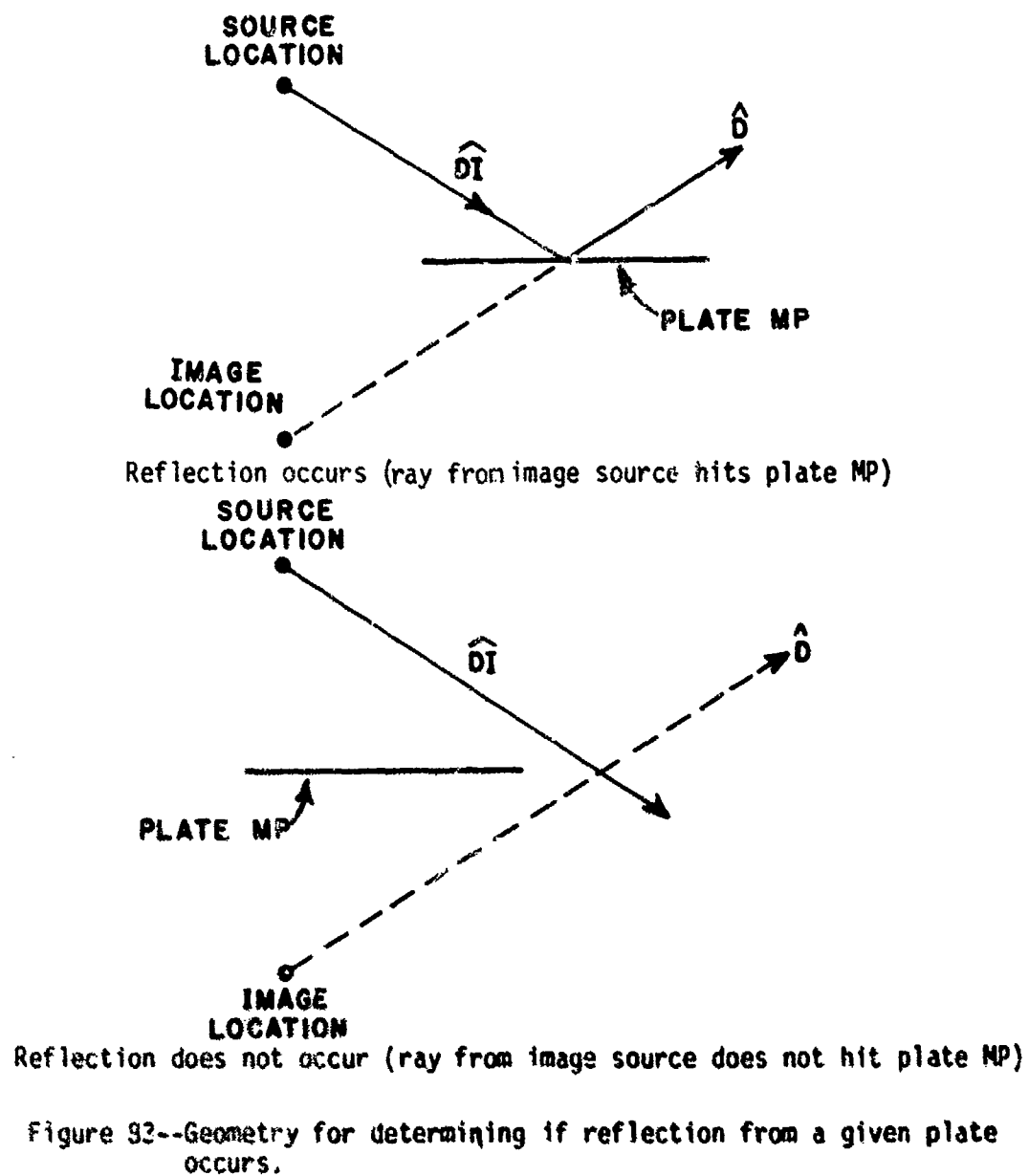
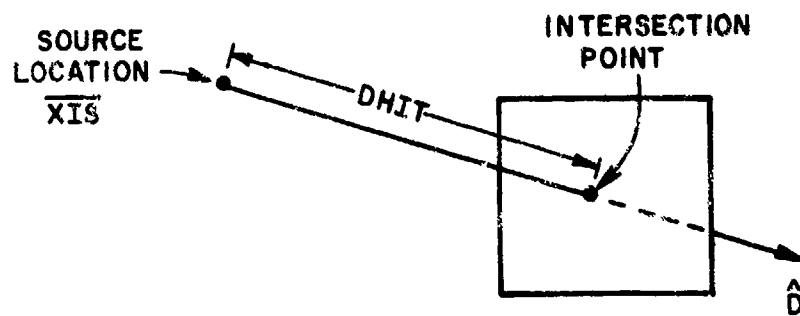
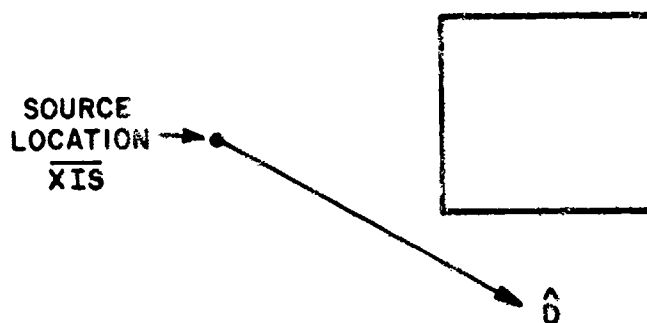


Figure 93--Geometry for determining if reflection from a given plate occurs.

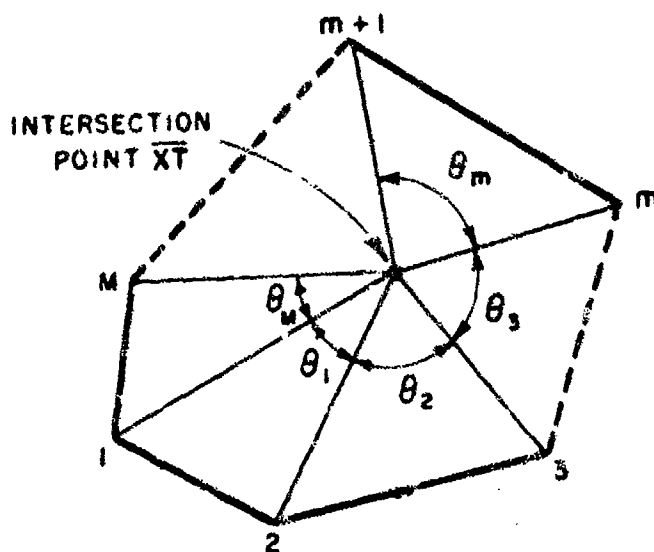


RAY HITS PLATE, LHIT = .TRUE.

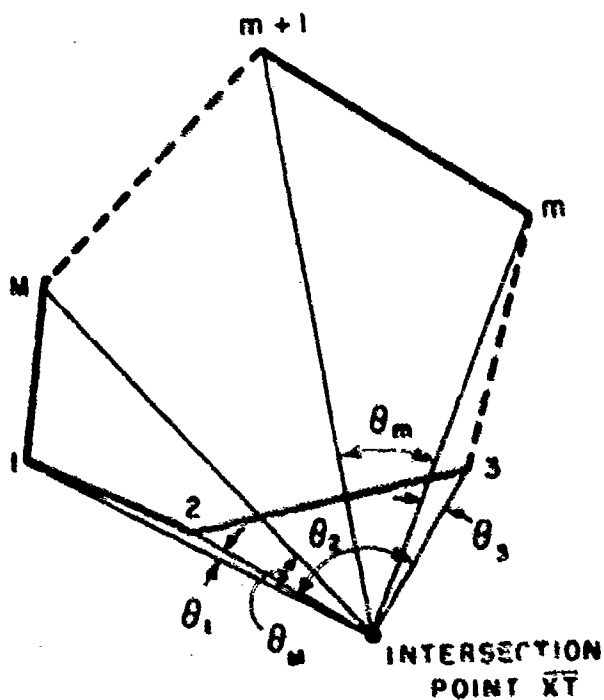


RAY DOES NOT HIT PLATE, LHIT = .FALSE.

Figure 34--Geometry for determining if a ray does or does not hit plate.



(a) RAY HITS PLATE



(b) RAY DOES NOT HIT PLATE

Figure 85--Geometry for deciding whether ray which hits plate plane hits finite plate.

METHOD

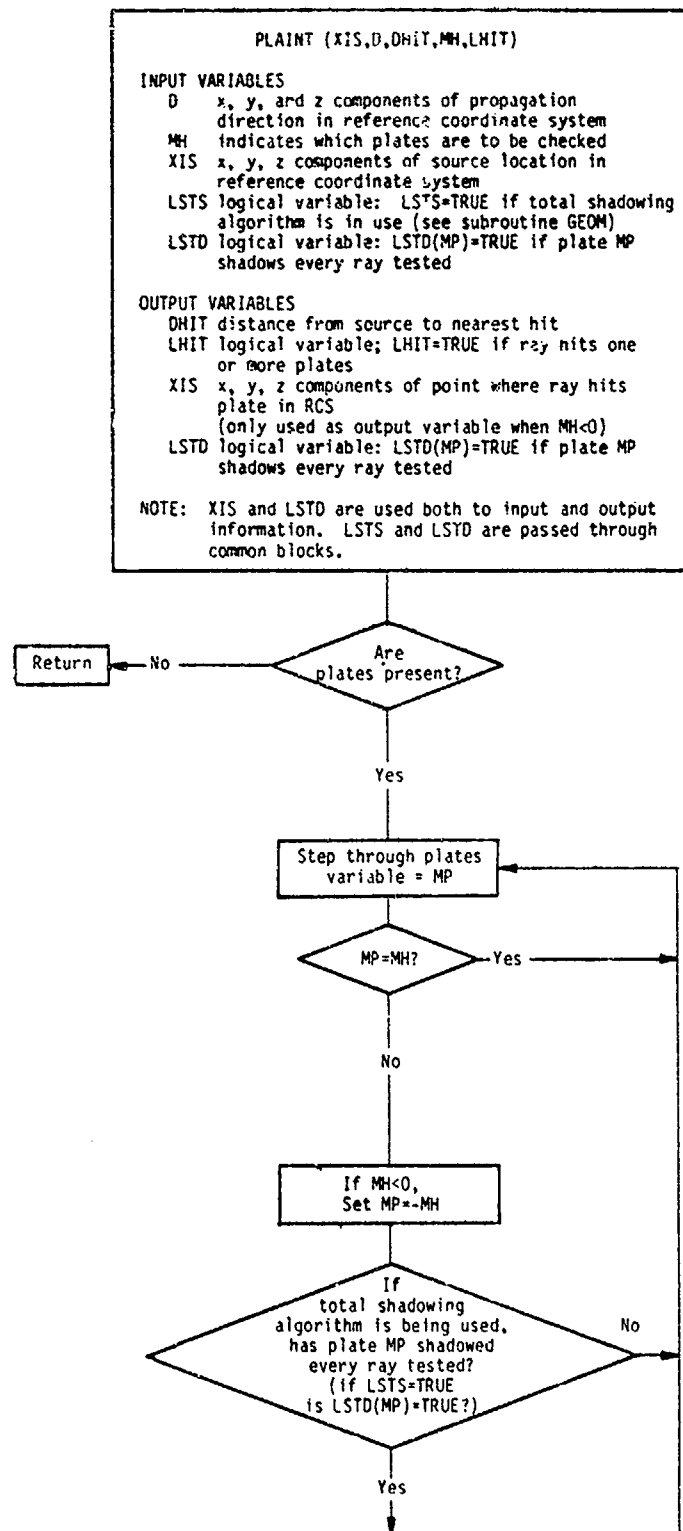
This subroutine is used for a number of functions:

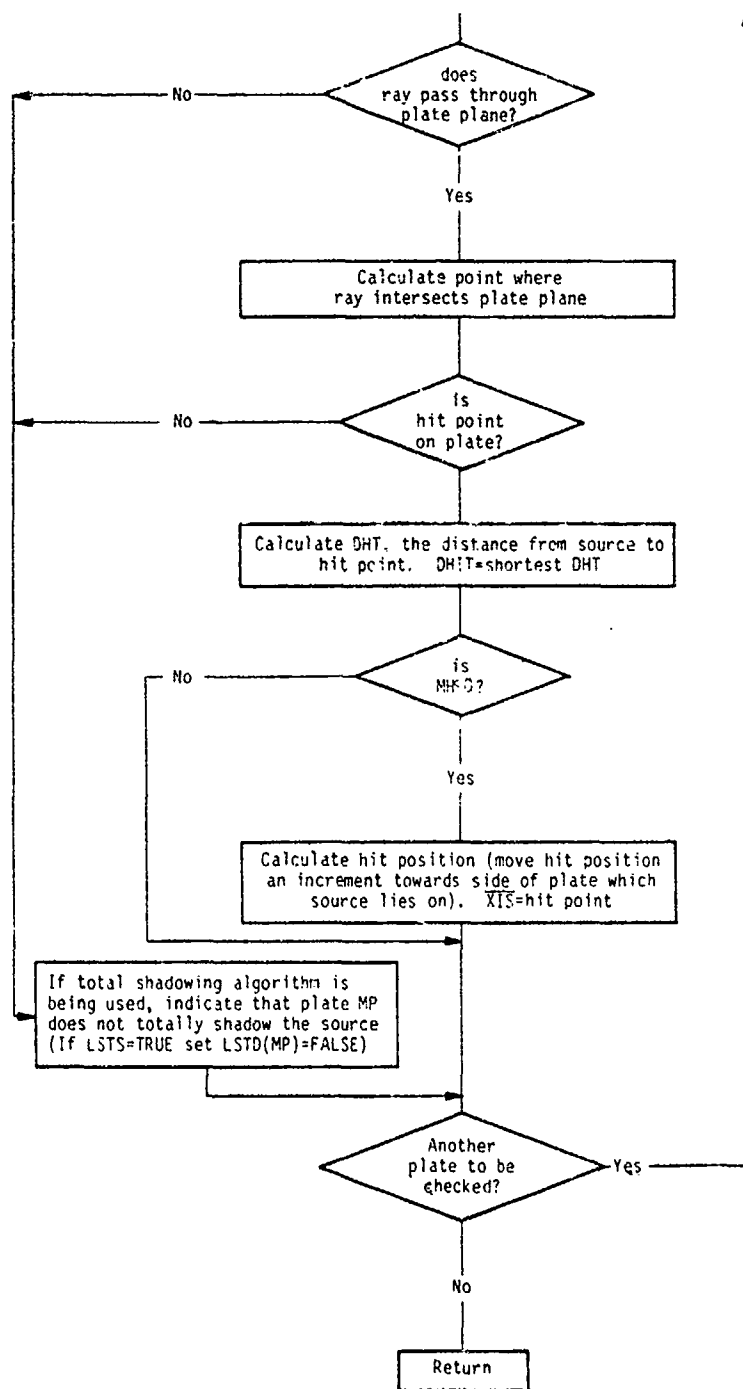
1. To determine if a source ray reflection from plate MP occurs. If a ray traveling from the source image location in the reflected ray direction passes through plate MP, the reflection will occur (see Figure 83). The routine only checks plate MP (set MH=MP). Note that the hit point (which is returned through the subroutine window) is the reflection point, and is used in shadowing tests.
2. To test to see if a ray is shadowed between scatter points (or between the source and a scatter point). The routine checks all plates (set MH=0) and records the distance from the first scatter (or source) position to the nearest hit (if the ray hits any of the plates). If the distance to the nearest hit is shorter than the distance between scatter points (or between the source and scatter point), the ray is shadowed, and the GTD term being computed is set to zero. Otherwise, the ray is not disturbed and computations are carried out. Note that if the first scatter point is a reflection or diffraction point on a plate, all plates except that plate are checked (set MH=MP).
3. To determine if ray after final scatter point (or source ray) is shadowed. If the final scatter point is a cylinder (or if the source field is being computed) all plates are checked. If the final scatter point is on plate MP, all plates except plate MP are checked. If the ray hits a plate (LHIT=TRUE) the ray is shadowed and the GTD term is set to zero. If LHIT=FALSE, the ray is not shadowed and propagates undisturbed.
4. To determine if any one plate totally shadows plate MP from the source (referred to as the "total shadowing algorithm"). The routine checks all plates except plate MP (set MH=MP) and remembers plates which shadow the ray every time the routine is called (see section 6 of subroutine GEOM). The total shadowing algorithm is activated when LSTS is set TRUE.

The hit algorithm first tests to see if a ray in the scatter direction will intersect the plane which the plate lies in by comparing the signs of the dot product of the scatter direction and the plate normal and the dot product of the vector from the source to a corner of the plate and the plate normal. If a hit is possible the intersection point on the plate plane is determined. Whether the intersection

point lies within the bounds of the plate is tested by summing the angles formed by the vectors from the intersection point to the various corners of the plate as shown in Figure 85. If the sum is zero the intersection point does not fall within the bounds of the plate. If the sum is 2π , the intersection point does fall within the bounds and the ray hits the plate. (See pp. 38-41, Reference 1).

FLOW DIAGRAM





SYMBOL DICTIONARY

AN DOT PRODUCT OF VECTOR FROM EDGE 1 OF PLATE MP TO
 SOURCE AND PLATE UNIT NORMAL
 CP COMPUTATIONAL VARIABLE
 D X, Y, AND Z COMPONENTS OF PROPAGATION DIRECTION
 IN REFERENCE COORDINATE SYSTEM
 DBI COMPUTATIONAL VARIABLE
 DBT COMPUTATIONAL VARIABLE
 DHIT DISTANCE FROM SOURCE TO NEAREST HIT
 DHT DISTANCE FROM SOURCE TO HIT POINT
 DN DOT PRODUCT OF PROPAGATION DIRECTION UNIT VECTOR
 AND PLATE UNIT NORMAL
 LHIT LOGICAL VARIABLE (SET TRUE IF RAY HITS AT LEAST
 ONE PLATE)
 LSTD SET TRUE IF PLATE MP TOTALLY SHADOWS PLATE MH
 FROM THE SOURCE
 LSTS SET TRUE IF TOTAL SHADOWING ROUTINE IS BEING USED
 ME DO LOOP VARIABLE
 MEX NUMBER OF EDGES ON PLATE MP
 MH SHOWS WHICH PLATES ARE TO BE CHECKED:
 MH=MP ONLY PLATE MP IS CHECKED
 MH=0 ALL PLATES ARE CHECKED
 MH=MP ALL PLATES EXCEPT MP ARE CHECKED
 MP INDEX VARIABLE (NUMBER OF PLATE BEING CHECKED)
 MPH INDEX VARIABLE
 MPP DO LOOP VARIABLE
 N DO LOOP VARIABLE
 RD COMPUTATIONAL VARIABLE
 XIS X, Y, Z COMPONENTS OF SOURCE LOCATION IN REFERENCE
 COORDINATE SYSTEM (ENTERING ROUTINE)
 XT X, Y, Z COMPONENTS OF HIT POSITION (LEAVING ROUTINE)
 X Y, Z COMPONENTS OF POINT WHERE RAY INTERSECTS
 PLATE PLANE

CODE LISTING

```

1 C-----
2 SUBROUTINE PLAINT(XIS,D,DHIT,MH,LHIT)
3 C!!!
4 C!!! DOES RAY HIT PLATE. IF MH=0 ALL PLATES ARE CHECKED.
5 C!!! IF MH=MP THEN ONLY MP CHECKED AND SOURCE POSITION
6 C!!! MOVED TO HIT POSITION IF RAY HITS MP.
7 C!!! IF MH=MP, THEN ALL PLATES OTHER THAN MP ARE CHECKED.
8 C!!!
9 DIMENSION XIS(3),D(3),XT(3)
10 LOGICAL LHIT,LPLA,LCYL,LSTS,LSTD
11 LOGICAL LGRND,LDEBUG,LTEST
12 COMMON/TEST/LDEBUG,LTEST
13 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
14 2,MEP(14),MPX
15 COMMON/PIS/PI,TPI,DPR,RPD
16 COMMON/LPLCY/LPLA,LCYL
17 COMMON/LSHDP/LSTS,LSTD(14)
18 COMMON/HITPLT/MPH
19 COMMON/GROUND/LGRND,MPXR
20 LHIT=.FALSE.
21 DHIT=0.
22 IF(.NOT.LPLA) RETURN
23 C!!! STEP THRU PLATES
24 DO 60 MPP=1,MPXR
25 MP=MPP
26 IF(MP.EQ.MH) GO TO 50
27 IF(MH.L1.0) MP=IABS(MH)
28 C!!! IF TOTAL SHADOWING ALGORITHM IS BEING USED, HAS PLATE MP
29 C!!! SHADOWED EVERY RAY TESTED?
30 IF(LSTS.AND..NOT.LSTD(MP)) GO TO 60
31 MEX=MEP(MP)
32 AN=0.
33 DO 5 N=1,3
34 5 AN=AN+(XIS(N)-X(MP,1,N))*VN(MP,N)
35 DN=D(1)*VN(MP,1)+D(2)*VN(MP,2)+D(3)*VN(MP,3)
36 C!!! DOES RAY PASS THRU PLATE PLANE?
37 IF(AN*DN.GE.0.) GO TO 50
38 DO 10 N=1,3
39 C!!! CALCULATE POINT WHERE RAY INTERSECTS PLATE PLANE
40 10 XT(N)=XIS(N)-AN*D(N)/DN
41 IF(MP.EQ.MPXR.AND.LGRND) GO TO 11
42 DBT=0.
43 C!!! IS HIT POINT ON PLATE?
44 DO 30 ME=1,MEX
45 MME=ME+1
46 IF(MME.GT.MEX) MME=1
47 RD=0.
48 DO 20 N=1,3
49 20 RD=RD+(X(MP,ME,N)-XT(N))*(X(MP,MME,N)-XT(N))
50 CP=VN(MP,1)*((X(MP,ME,2)-XT(2))*(X(MP,MME,3)-XT(3))
51 2-(X(MP,ME,3)-XT(3))*(X(MP,MME,2)-XT(2)))
52 CP=CP+VN(MP,2)*((X(MP,ME,3)-XT(3))*(X(MP,MME,1)-XT(1))
53 2-(X(MP,ME,1)-XT(1))*(X(MP,MME,3)-XT(3)))
54 CP=CP+VN(MP,3)*((X(MP,ME,1)-XT(1))*(X(MP,MME,2)-XT(2))
55 2-(X(MP,ME,2)-XT(2))*(X(MP,MME,1)-XT(1)))
56 DBI=BTAN2(CP,RD)
57 DHT=DBT+DBI
58 30 CONTINUE
59 IF(ABS(DBT).LT.PI) GO TO 50
60 C!!! CALCULATE DISTANCE TO HIT (DHIT=SHORTEST DHT)
61 11 DHT=0.
62 DO 40 N=1,3
63 40 DHT=DHT+(X1(N)-XIS(N))*(XT(N)-XIS(N))
64 DHT=SQRT(DHT)+1.E-5
65 IF(LHIT.AND.(DHT.GT.DHIT)) GO TO 60

```

```

66      LHIT=.TRUE.
67      DHIT=DH1
68      MPH=MP
69      IF(MH.GE.0) GO TO 60
70      DO 45 N=1,3
71 C!!!  MOVE HIT POSITION AN INCREMENT TOWARDS SIDE OF PLATE
72 C!!!  WHICH SOURCE LIES ON
73 45     XIS(N)=XT(N)-SIGN(1.E-5,AN)*VN(MP,N)
74      GO TO 61
75 50     CONTINUE
76      IF(MH.LT.0) GO TO 61
77 C!!!  IF TOTAL SHADOWING ROUTINE IS BEING USED, INDICATE
78 C!!!  THAT PLATE MP DOES NOT SHADOW SOURCE
79      IF(LSTS) LSTD(MP)=.FALSE.
80 60     CONTINUE
81 61     IF (.NOT.LTEST) GO TO 62
82      WRITE (6,63)
83 63     FORMAT (/,' TESTING PLAINT SUBROUTINE')
84      WRITE (6,*) XIS
85      WRITE (6,*) D
86      WRITE (6,*) DHIT,MH,LHIT
87 62     RETURN
88      END

```

POLYRT

PURPOSE

To solve an Mth order polynomial equation.

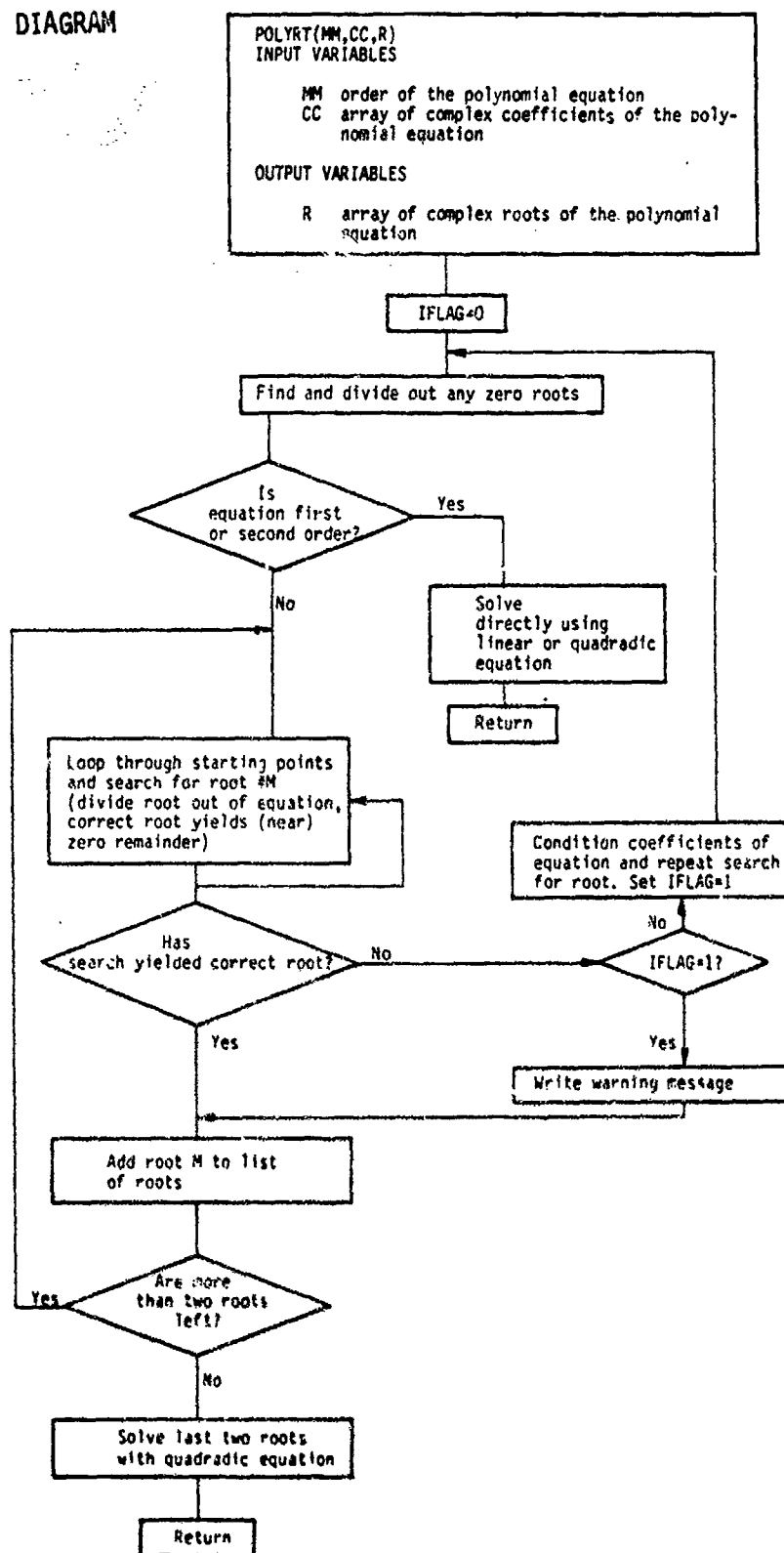
METHOD

This subroutine solves for the roots of an Mth order polynomial,

$$C_M Z^M + C_{M-1} Z^{M-1} + \cdots C_1 Z^1 + C_0 = 0.$$

The roots of the polynomial are found using the Newton-Raphson method of iterated synthetic division [16]. The coefficients are stored such that $C_M = CC(M+1)$, $C_0 = CC(1)$, etc.

FLOW DIAGRAM



SYMBOL DICTIONARY

C WORKING ARRAY OF POLYNOMIAL COEFFICIENTS
 CC A COMPLEX ARRAY CONTAINING THE POLYNOMIAL COEFFICIENTS
 CMAX MAGNITUDE OF LARGEST COEFFICIENT
 CNEW ARRAY CONTAINING COEFFICIENTS OF POLYNOMIAL LEFT
 AFTER THE PROSPECTIVE ROOT HAS BEEN FACTORED OUT
 CNNW ARRAY CONTAINING COEFFICIENTS OF POLYNOMIAL LEFT AFTER
 THE PROSPECTIVE ROOT HAS BEEN FACTORED OUT TWICE
 EPS SMALL NUMBER (RELATIVE TO LARGEST COEFFICIENT)
 ICONJ INDEX FOR TRYING THE CONJUGATE OF THE PREVIOUS ROOT
 AS A GUESS
 ICCUNT INDEX ON THE NUMBER OF TIMES THE ITERATION PROCEDURE
 SEARCHES FOR A ROOT
 IFLAG FLAG USED TO INDICATE IF ALL POSSIBLE STARTING VALUES
 HAVE BEEN TRIED
 ISTART INDEX FOR STARTING VALUES
 LIMIT MAXIMUM NUMBER OF ITERATIONS USED TO SEARCH FOR THE ROOT
 M ORDER OF POLYNOMIAL BEING WORKED ON
 MI COMPUTATIONAL VARIABLE

 MM ORDER OF THE EQUATION
 MMPI MM PLUS ONE
 MN ORDER OF ONCE FACTORED POLYNOMIAL BEING WORKED ON
 O MAGNITUDE OF POLYNOMIAL COEFFICIENTS
 R A COMPLEX ARRAY CONTAINING THE ROOTS OF THE EQUATION
 RJ REMAINDER LEFT AFTER PROSPECTIVE ROOT HAS BEEN
 FACTORED OUT
 RJP REMAINDER LEFT AFTER PROSPECTIVE ROOT HAS BEEN FACTORED
 OUT TWICE
 RT PROSPECTIVE ROOT BEING ITERATED
 SR SQUARE ROOT OF $(C(2)*C(2)-4*C(1)*C(3))$
 START ARRAY CONTAINING INITIAL GUESS OF ROOT LOCATIONS
 TEST BOUND USED TO DETERMINE IF THE PROSPECTIVE ROOT
 HAS CONVERGED
 XI IMAGINARY PART OF CC
 XR REAL PART OF CC

CODE LISTING

```

1 C-----
2      SUBROUTINE PCLYRT(M,CC,R)
3 C!!!
4 C!!! THIS ROUTINE SOLVES A COMPLEX POLYNOMIAL EQUATION.
5 C!!! MM IS THE ORDER OF THE EQUATION
6 C!!! CC IS A COMPLEX ARRAY CONTAINING THE COEFFICIENTS.
7 C!!! CC(1) IS THE CONSTANT TERM, CC(2) THE COEFFICIENT OF Z,
8 C!!! -----,CC(MM+1) THE COEFFICIENT OF Z**MM.
9 C!!! R IS A COMPLEX ARRAY IN WHICH THE ROOTS WILL BE RETURNED.
10 C!!! IN THE DATA STATEMENT LIMIT IS THE NUMBER OF CYCLES
11 C!!! WHICH WILL BE ALLOWED BEFORE THE SEARCH FOR A
12 C!!! PARTICULAR ROOT IS TERMINATED. TEST IS THE MAXIMUM
13 C!!! INEQUALITY OF THE EQUATION ALLOWED BEFORE A ROOT IS
14 C!!! ACCEPTED.
15 C!!!
16      COMPLEX C(21),CC(21),CNEW(21),R(20),SR,RT,Y,DY,RTP
17      COMPLEX START(4),CNNH(21),RJ,RJP
18      DATA START/(1.,1.), (1.,0.), (-1.,-1.), (-1.,0.)/
19      DATA TEST,LIMIT/1.E-05,100/
20 C!!! COPY THE INPUT PARAMETERS CC AND MM INTO C AND M.
21      IFLAG=0
22      MMP1=MM+1
23      CMAX=BAES(CC(1))
24      DO 9 I=1,MMP1
25          C(I)=CC(I)
26          IF(BABS(CC(1)).GT.CMAX) CMAX=BABS(CC(1))
27 9      CONTINUE
28      EPS=1.E-5*CMAX
29 3535 M=MM
30      ICONJ=0
31 C!!! FIND AND DIVIDE OUT ANY ZERO ROOTS.
32 2      Q=BABS(C(M+1))
33          IF(Q.LT.EPS) GO TO 7
34          Q=BABS(C(1))
35          IF(Q.GT.EPS) GO TO 1
36          DO 8 I=1,M
37              C(I)=C(I+1)
38              R(M)=(0.,0.)
39              M=M-1
40          IF(M.NE.0) GO TO 2
41      RETURN
42 1      DO 3 N=1,M
43          C(N)=C(N)/C(M+1)
44          C(M+1)=(1.,0.)
45 C!!! IF EQUATION IS 1ST OR 2ND ORDER SOLVE DIRECTLY AND RETURN.
46          IF(M-2) 5,6,4
47 5      R(1)=C(1)
48      RETURN
49 C!!! START SEARCH FOR A ROOT.
50 4      DO 14J 1START=1,4
51 24      RT=START/(1START)
52          IF(ICONJ.EQ.1) RT=CONJG(R(M+1))
53          ICOUNT=0
54 14      CNEW(M)=(1.,0.)
55          MN=M-1
56          CNNH(MN)=(1.,0.)
57          ICOUNT=ICOUNT+1
58          IF(ICOUNT.GT.LIMIT) GO TO 141
59          DO 111 I=2,M
60              M1=M-1
61 111      CNEW(M1+1)=C(M1+2)+RT*CNEW(M1-2)
62              R1=C(1)+RT*CNEW(1)
63              C=BABS(R1)
64              IF(C.LE.TEST) GO TO 12
65              DO 112 I=2,MN
66                  M1=M1-1

```

```

07 112 CNW(MI+1)=CNEW(MI+2)+RT*CNW(MI+2)
08 HJP=CNEK(1)+RT*CNW(1)
09 RT=RT-RJ/RJP
10 GO TO 14
11 141 CONTINUE
12 IF(ICONJ.NE.1) GO TO 140
13 ICONJ=0
14 GO TO 24
15 140 CONTINUE
16 IF(IFLAG.EQ.1) GO TO 15
17 IFLAG=1
18 DO 9090 JJ=1,MPI
19 XR=REAL(CC(JJ))
20 XI=AIMAG(CC(JJ))
21 IF(ABS(XR).LT.EPS) XR=0.
22 IF(ABS(XI).LT.EPS) XI=0.
23 C(JJ)=CMPLX(XR,XI)
24 9090 CONTINUE
25 GO TO 3535
26 15 WRITE(6,16) N,0
27 16 FORMAT(1H0,40H CYCLE LIMIT EXCEEDED WHILE FINDING ROOT,13,
28 218H FINAL INEQUALITY ,F10.4)
29 12 CONTINUE
30 DO 18 I=1,M
31 18 C(I)=CNEK(I)
32 R(M)=RT
33 M=M-1
34 ICONJ=ICONJ+1
35 IF(ICONJ.GT.1) ICONJ=0
36 C!!! IF MORE THAN TWO ROOTS LEFT RECYCLE THE SEARCH.
37 IF(M.GT.2) GO TO 4
38 C!!! FIND THE LAST TWO ROOTS BY THE QUADRATIC FORMULA.
39 C SR=CSQRT(C(2)*C(2)-4.0*C(1)*C(3))
100 R(1)=(-C(2)+SR)*0.5/C(3)
101 R(2)=(-C(2)-SR)*0.5/C(3)
102 RETURN
103 END

```

PRIOUT

PURPOSE

To output field data in standard format: 4 integer indicators and then magnitude and phase of E-theta and E-phi components.

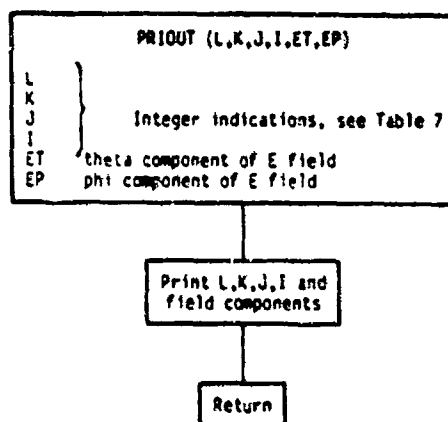
METHOD

This subroutine is activated by setting LOUT=.TRUE. When the individual field components are being printed out, that is when $L \neq 1000$ and when $L \neq K$ and $J \neq I$, only the fields with $|ET| > 0$ or $|EP| > 0$ are printed out. A list of the different indicator numbers and what field they correspond to is given in Table 7.

Table 7
Individual field types printed when LOUT=.TRUE.

L	K	J	I	Field Type
100	0	0	0	Direct field when plates are present
200	MP	0	0	Field reflected from plate MP
300	MP	MPP	0	Field reflected from plate MP then reflected from plate MPP
600	MP	ME	0	Field diffracted from edge ME of plate MP
650	MP	ME	0	Field diffracted from the corners of edge ME of plate MP
700	MR	MP	ME	Field reflected from plate MR then diffracted from edge ME of plate MP
750	MR	MP	ME	Field reflected from plate MR then diffracted by the corners of edge ME of plate MP
800	MP	ME	MR	Field diffracted from edge ME of plate MP then reflected from plate MR
850	MP	ME	MR	Field diffracted from the corners of edge ME of plate MP then reflected from plate MR
110	0	0	0	Direct field when only cylinders alone are present
120	0	0	0	Geometrical optics field reflected by cylinder (for comparison only)
130	0	0	0	Field scattered by the curved surface of the cylinder
150	MC	0	0	Field reflected by end cap MC of the cylinder
500	MC	0	0	Field diffracted by the end cap rim MC of the cylinder
240	MP	0	0	Geometrical optics field reflected from plate MP then reflected from the curved surface of the cylinder. (For comparison only)
250	MP	0	0	Field reflected from plate MP and then scattered by the curved surface of the cylinder
410	MP	0	0	Geometrical optics field reflected from the curved surface of the cylinder and then reflected from plate MP. (For comparison only)
420	MP	0	0	Field scattered from the curved surface of the cylinder then reflected from plate MP
540	MP	ME	0	Field reflected from the curved surface of the cylinder then diffracted by edge ME of plate MP
950	MP	ME	0	Field diffracted from edge ME of plate MP then reflected from the curved surface of the cylinder
1ANGLE	1ANGLE	1ANGLE	1ANGLE	Sum of fields of a given type (1ANGLE) for a given angle (1ANGLE)
1000	1ANGLE	1ANGLE	1ANGLE	Total field for a given angle (1ANGLE)

FLOW DIAGRAM



CODE LISTING

```

1 C-----
2 SUBROUTINE PRIOUT(L,K,J,I,ET,EP)
3 C!!!
4 C!!! PRINT OUT DATA IN STANDARD FORMAT.
5 C!!! INTEGER INDICATORS, THEN MAG. AND PHASE
6 C!!! OF E-THETA&PHI COMPONENTS.
7 C!!!
8 COMPLEX ET,EP
9 COMMON/FIS/PI,TPI,OPR,RPO
10 UTH=DABS(ET)
11 UTP=OPR*BTAN2(ATHAG(ET),REAL(ET))
12 UPH=UABS(EP)
13 UPP=OPR*BTAN2(ATHAG(EP),REAL(EP))
14 IF(L.EQ.1)GO TO 2
15 IF(L.EQ.2)AND(J.EQ.1)GO TO 2
16 IF(UTH.LT.1.E-5)AND(UPP.LT.1.E-5)RETURN
17 2 WRITE(6,1) L,K,J,I,UTH,UTP,UPH,UPP
18 1 FORMAT(1H ,4I5,2F15.6,5X,2F15.6)
19 RETURN
20 END
  
```

QFUN

PURPOSE

To compute the q^* function for the cylinder's acoustically hard diffraction coefficient.

METHOD

The q^* function is defined as [14,15]

$$q^*(x) = \frac{1}{2\sqrt{\pi x}} + \hat{p}_h(x) e^{j\pi/4}$$

where

$$\hat{p}_h(x) = \frac{e^{-j\pi/4}}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{QV(\tau)}{Qw_2(\tau)} e^{-jx\tau} d\tau$$

and

$V(\tau)$ and $w_2(\tau)$ are Fock type Airy functions,

and

$$Q = \frac{\partial}{\partial \tau}.$$

The q^* function is computed as follows:

1) for $x \leq -3$

$$q^*(x) = \frac{1}{2\sqrt{\pi x}} - \frac{1}{2}\sqrt{|x|} \left(1 - j\frac{2}{x^3}\right) e^{j\frac{x^3}{12}} e^{j\pi/4},$$

2) for $-3 < x < 2$

$$q^*(x) = q^*(x_i) + \frac{(x-x_i)}{(x_{i+1}-x_i)} (q^*(x_{i+1}) - q^*(x_i)),$$

where the $q^*(x_i)$ are tabulated values [14,15] and $x_{i+1} - x_i = 0.1$ with $x_i < x < x_{i+1}$.

3) for $x \geq 2$

$$q^*(x) = \frac{1}{2\sqrt{\pi x}} - \frac{e^{j\pi/6}}{2\sqrt{\pi}} \sum_{n=1}^5 \frac{e^{x\bar{q}_n} e^{-j\frac{5\pi}{6}}}{\bar{q}_n [A_i(-\bar{q}_n)]}$$

where $A_i(r)$ is the Miller type Airy function.

SYMBOL DICTIONARY

AMC $-0.5 * \text{CEXP}(J * \text{PI} / 6) / \text{SQRT}(\text{PI})$
 AQ MILLER TYPE AIRY FUNCTION AT Q
 C $0.5 / \text{SQRT}(\text{PI})$
 EXC $\text{CEXP}(-5 * \text{PI} / 6)$
 I SMALLEST INTEGER CLOSEST TO $10 * X$
 Q ZERGES OF DERIVATIVE OF MILLER TYPE AIRY FUNCTION
 QFUN Q FUNCTION
 QI IMAGINARY PART OF TABULATED Q FUNCTION
 QR REAL PART OF TABULATED Q FUNCTION
 X ARGUMENT OF Q FUNCTION
 XI REAL NUMBER REPRESENTATION OF I

CODE LISTING

```

1 C-----
2      COMPLEX FUNCTION QFUN(X)
3 C!!!
4 C!!! COMPUTES THE Q FUNCTION OF THE CYLINDER'S
5 C!!! DIFFRACTION COEFFICIENT (HARD CASE)
6 C!!!
7      DIMENSION QR(61),QI(61)
8      COMPLEX AMC,EXC
9      DIMENSION Q(5),AQ(5)
10     COMMON/PIS/PI,TPI,DPR,RPD
11     DATA AMC,EXC/(-0.24430,-0.14105),(-0.866025,-0.5)/
12     DATA C/0.28209/
13     DATA Q/1.01879,3.24820,4.82010,6.16331,7.37218/
14     DATA AQ/0.53566,-0.41942,0.38041,-0.35791,0.34230/
15     DATA QR/-.229,-.411,-.559,-.673,-.754,-.807,-.834,-.841
16     2,-.852,-.810,-.780,-.744,-.705,-.665,-.625,-.587,-.551,
17     2,-.517,-.486,-.458,-.432,-.409,-.388,-.369,-.352,-.335,
18     2,-.320,-.306,-.293,-.279,-.266,-.253,-.239,-.226,-.212,
19     2,-.198,-.184,-.170,-.155,-.141,-.126,-.112,-.098,-.084,
20     2,-.071,-.058,-.046,-.034,-.023,-.012,-.0026,.0064,.015,
21     2,.022,.029,.030,.041,.046,.051,.056,.061/
22     DATA QI/-.838,-.771,-.676,-.562,-.440,-.317,-.199,-.090
23     2,-.008,.094,.166,.226,.274,.311,.338,.357,.368,.372,.371
24     2,.365,.356,.342,.327,.309,.289,.268,.246,.223,.200,.177
25     2,.154,.131,.109,.088,.067,.048,.031,.014,-.0013,-.015
26     2,-.027,-.038,-.048,-.056,-.062,-.068,-.072,-.075,-.078
27     2,-.079,-.079,-.079,-.078,-.077,-.075,-.072,-.070,-.067
28     2,-.064,-.061,-.059/
29     IF(X.LE.-3.)GO TO 1
30     IF(X.GE.2.)GO TO 2
31     I=((3.+X)*10.)
32     XI=FLOAT(I)-30.
33     I=I+1
34     QFUN=CMPLX(QR(I),-QI(I))+(10.*X-XI)*CMPLX(QR(I+1)-QR(I),
35     2-QI(I+1)+QI(I))
36     RETURN
37 1    QFUN=.5*(1./SQRT(PI)*X)-SQRT(ABS(X))*CEXP(CMPLX(0.,0.25
38     2*PI*X*X*X/12.))*CMPLX(1.,-2./(X*X*X))
39     RETURN
40 2    QFUN=(0.,0.)
41     DO 3 N=1,5
42 3    QFUN=QFUN+CEXP(Q(N)*X*EXC)/AQ(N)/AQ(N)/Q(N)
43     QFUN=QFUN*AMC+C/X
44     RETURN
45     END
  
```

RADCV

PURPOSE

To compute the longitudinal and transverse radii of curvature of the elliptic cylinder at a given point.

METHOD

The longitudinal radius of curvature of the elliptic cylinder (in the plane of incidence) at the point defined by elliptical angle VR is given by

$$\rho_g = \frac{(A^2 \sin^2 VR + B^2 \cos^2 VR)^{3/2}}{AB \sin^2 \alpha_s}.$$

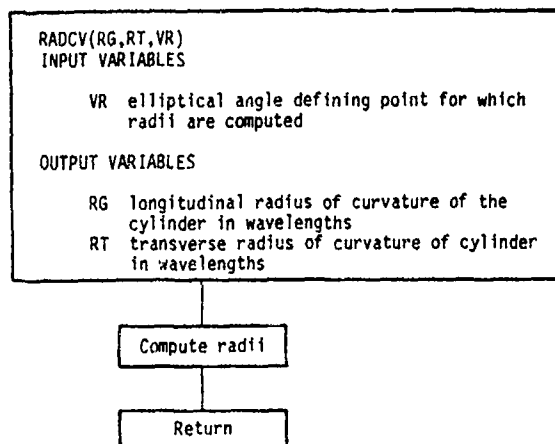
The transverse radius of curvature at the point defined by ell. angle VR is given by

$$\rho_t = \frac{(A^2 \sin^2 VR + B^2 \cos^2 VR)^{3/2}}{AB \sin^2(\alpha_s - \pi/2)},$$

where

$$\begin{aligned}\alpha_s &= AS \\ \rho_g &= RG \\ \rho_t &= RT.\end{aligned}$$

FLOW DIAGRAM



SYMBOL DICTIONARY

RG	RADIUS OF CURVATURE IN THE PLANE OF INCIDENCE
RG1	RADIUS OF CURVATURE OF THE ELLIPTIC CYLINDER IN THE PRINCIPAL (X-Y) PLANE
RT	RADIUS OF CURVATURE TRANSVERSE TO THE PLANE OF INCIDENCE
VR	ELLIPTIC ANGLE DEFINING THE DESIRED POINT ON CYLINDER

CODE LISTING

```

1 C-----
2 SUBROUTINE RADCY(RG,RT,VR)
3 C!!!
4 C!!! COMPUTES RADII OF CURVATURE OF ELLIPTIC CYLINDER
5 C!!!
6 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
7 COMMON/GTD/AS,ID,SAS,SASP,CAS
8 DN=SQRT(A*A*SIN(VR)*SIN(VR)+B*B*COS(VR)*COS(VR))
9 RGT=DN*DN*DN/A/B
10 RG=RG1/SAS/SAS
11 IF(SASP.LT.1.E-5)GO TO 1
12 RT=RGT/SASP/SASP
13 RETURN
14 1 RT=1.E20
15 RETURN
16 END
  
```

RCLDPL

PURPOSE

To compute the far-zone electric field for a source ray which is reflected by the elliptic cylinder and then diffracted by a given edge on a given plate.

PERTINENT GEOMETRY

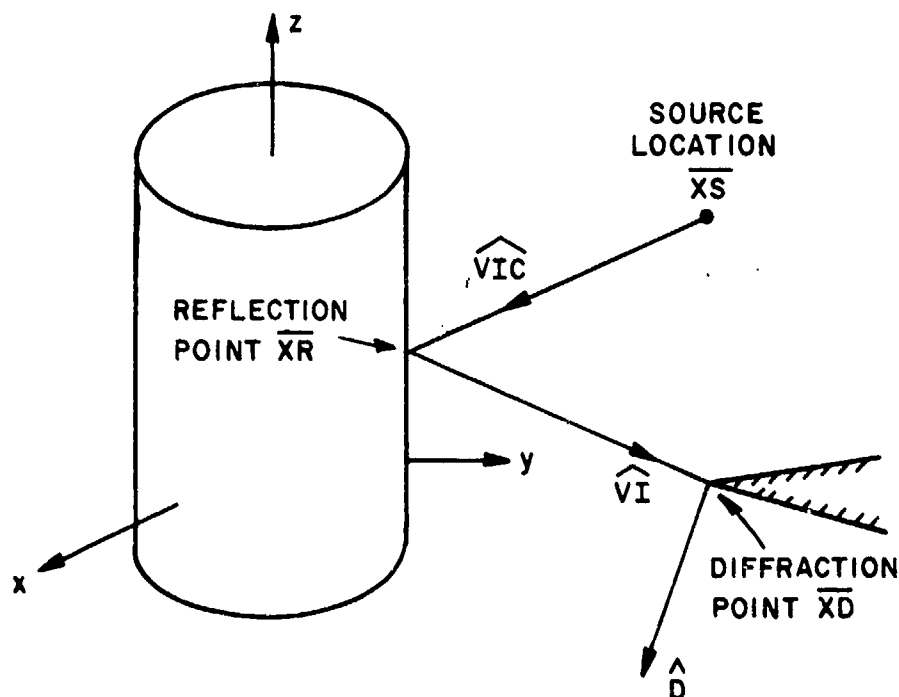


Figure 86--Ray reflected by cylinder and then diffracted by plate edge.

METHOD

The field reflected by the elliptic cylinder and then diffracted by a plate edge is calculated in this subroutine. The field reflected by the cylinder is found using geometrical optics[4]. This causes an astigmatic tube of rays to be incident on the plate edge. The uniform Geometrical Theory of Diffraction[4] is then used to find the diffracted field from the edge. The resultant field in the far zone has the form (pp. 154-155, Reference 1)

$$\vec{E}^{r,d} = \vec{E}^i(Q_R) \cdot \vec{R} \cdot \vec{D} \sqrt{\frac{\rho_1^r \rho_2^r}{(\rho_1^r + s')(\rho_2^r + s')}} \sqrt{\rho_e^i} e^{-jks'} \frac{e^{-jks}}{s} ,$$

where $\vec{E}^i(Q_R)$ is the incident field at the reflection point Q_R , $\vec{R}$ is the dyadic reflection coefficient, $\vec{D}$ is the dyadic edge diffraction coefficient, ρ_1^r and ρ_2^r are the reflected ray caustic distances, ρ_e^i is the incident caustic distance on the edge, s' is the distance from the reflection point to the diffraction point, and s is the distance from the diffraction point in the far zone. The geometry is shown in Figure 85 and further illustrations can be found in the write ups for subroutines REFCYL and DIFPLT. The phase of the field is referred to the reference coordinate system origin so that

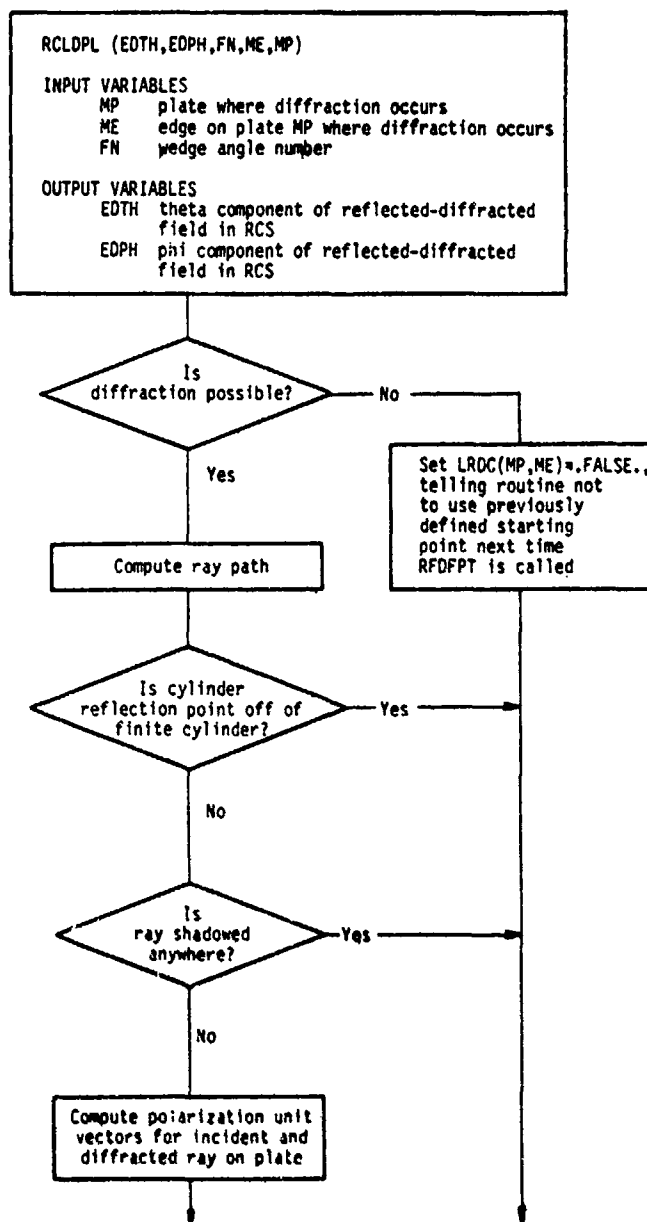
$$\frac{e^{-jks}}{s} = e^{jk\hat{D} \cdot \vec{X}_d} \frac{e^{-jkR}}{R} .$$

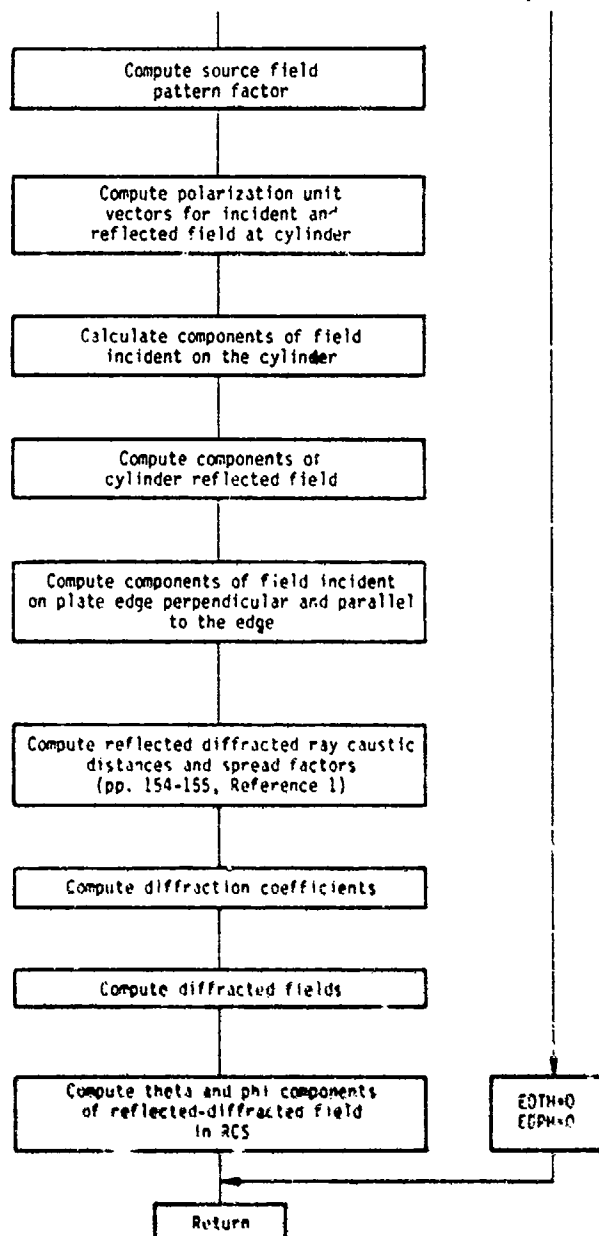
The reflected-diffracted field then has the form

$$\vec{E}^{r,d} = W_m (EDTH\hat{\theta} + EDPH\hat{\phi}) \frac{e^{-jkR}}{R} ,$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM





SYMBOL DICTIONARY

BO	DIFFRACTED FIELD POLARIZATION UNIT VECTOR PARALLEL TO EDGE
BOP	INCIDENT FIELD POLARIZATION UNIT VECTOR PARALLEL TO EDGE
DD	NORMALIZATION CONSTANT FOR CYLINDER TANGENT VECTOR
DH	EDGE DIFFRACTION COEFFICIENT FOR HARD FIELD COMPS.
DHIT	DISTANCE FROM SOURCE TO HIT POINT (FROM PLAIN)
DOTP	TEST PARAMETER USED TO DETERMINE IF REFL IS LEGAL
DS	DIFFRACTION COEFFICIENT FOR SOFT FIELD COMPONENTS
DV	DOT PRODUCT OF EDGE UNIT VECTOR AND DIFFRACTED RAY PROPAGATION DIRECTION
EDPH	PHI COMPONENT OF DIFFRACTED FIELD IN RCS
EDPL	DIFFRACTED FIELD COMPONENT PARALLEL TO EDGE
EDPR	DIFFRACTED FIELD COMPONENT PERPENDICULAR TO EDGE
EDTH	THETA COMPONENT OF DIFFRACTED FIELD IN RCS
EIPL	COMPONENT OF FIELD INCIDENT ON CYLINDER (OR PLATE) PARALLEL TO PLANE OF INCIDENCE (OR EDGE)
EIPR	COMPONENT OF FIELD INCIDENT ON CYLINDER (OR PLATE) PERPENDICULAR TO PLANE OF INCIDENCE (OR EDGE)
EIX	SOURCE PATTERN FACTORS FOR X,Y,Z COMPONENTS OF INCIDENT E-FIELD
EIY	
EIZ	
ERX	
ERY	X,Y,Z COMPONENTS OF CYLINDER REFLECTED FIELD IN RCS
ERZ	
EXPH	
LDRC	
LDRC	COMPLEX PHASE AND SPREADING FACTOR
LDRC	SET TRUE IF REFL DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE (OR FOR NEXT PATTERN ANGLE (WHEN LEAVING ROUTINE))
LHIT	SET TRUE IF RAY HITS PLATE (FROM PLAIN)
ME	EDGE ON PLATE MP WHERE DIFFRACTION OCCURS
MP	PLATE WHERE DIFFRACTION OCCURS
PH	DIFFRACTED FIELD POLARIZATION UNIT VECTOR NORMAL TO EDGE
PHICR	PHI COMPONENT OF FIELD INCIDENT ON CYLINDER IN RCS
PHC	INCIDENT FIELD POLARIZATION UNIT VECTOR NORMAL TO EDGE
PSOR	INCIDENT RAY PHI ANGLE IN DIFFRACTION POINT COORD SYS
PSR	DIFFRACTED RAY PHI ANGLE IN DIFFRACTION POINT COORD SYS
NH11	CAUSTIC DISTANCE OF CYLINDER REFLECTED FIELD INCIDENT ON EDGE IN THE DIRECTION PERPENDICULAR TO THE EDGE
NH12	CAUSTIC DISTANCE OF CYLINDER REFLECTED FIELD INCIDENT ON EDGE IN THE DIRECTION PARALLEL TO THE EDGE
NH1E	EDGE CAUSTIC DISTANCE
NH01	RAY SPREADING RADIUS AT CYLINDER IN PLANE NORMAL TO PLANE OF INCIDENCE
NH02	RAY SPREADING RADIUS AT CYL IN PLANE OF INCIDENCE
SNAG	LENGTH OF RAY FROM REFL POINT ON CYL TO SOURCE
SP	DISTANCE BETWEEN REFLECTION AND DIFFRACTION POINT
THICR	THETA COMPONENT OF INCIDENT RAY DIRECTION ON CYLINDER IN RCS
TPP	DISTANCE PARAMETER FOR EDGE DIFFRACTED FIELD
UIIPA	X,Y,Z COMPONENTS OF INCIDENT POLARIZATION UNIT VECTOR PARALLEL TO PLANE OF INCIDENCE
UIIPPY	
UIIPZ	X,Y,Z COMPONENTS OF INC/REFL POLARIZATION UNIT VECTOR PERPENDICULAR TO PLANE OF INCIDENCE
UIIPA	
UIIPPY	X,Y,Z COMPONENTS OF REFLECTED POLARIZATION UNIT VECTOR PARALLEL TO THE PLANE OF INCIDENCE
UIIPZ	
UIIPA	X,Y,Z COMPONENTS OF RAY PROPAGATION DIRECTION OF RAY INCIDENT ON DIFFRACTION POINT
UIIPPY	
UIIPZ	X,Y,Z COMPONENTS OF RAY PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER
UIIPA	

VH ELL ANGLE DEFINING REFLECTION POINT ON CYL (2-D)
XD X,Y,Z COMPONENTS OF DIFFRACTION POINT IN WCS
ADP MODIFIED DIFFRACTION POINT LOCATION FOR SHADOWING TEST
AH X,Y,Z COMPONENTS OF REFLECTION POINT ON CYL

CODE LISTING

```

1 C-----
2 SUBROUTINE RCLDPL(EDTH,EDPH,FN,ME,MP)
3 C!!!
4 C!!! COMPUTES THE FIELD REFLECTED FROM THE ELLIPTIC CYLINDER
5 C!!! THEN DIFFRACTED FROM EDGE #ME OF PLATE #MP
6 C!!!
7 COMPLEX EF,EG,EIPR,EIPL,EXPH,DS,DH,DPS,DPH,EDPR,EDPL,EDTH,EDPH
8 COMPLEX ERFR,ERPP,EIX,EIY,EIZ,ERX,ERY,ERZ
9 DIMENSION UN(2),UB(2),VIC(3),XR(3)
10 DIMENSION VI(3),XD(3),PHO(3),PH(3),ROP(3),RO(3),XDP(3)
11 LOGICAL LHIT,LRDC,LDEBUG,LTEST
12 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
13 2,MEP(14),MPX
14 COMMON/GEORINF/XS(3),VXS(3,3)
15 COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
16 COMMON/GEONEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
17 COMMON/ENDHCL/VCD(14,6),UCD(14,6),BCD(14,6,2)
18 COMMON/THPHUV/DT(3),DP(2)
19 COMMON/FIS/PI,TPI,DPR,RHD
20 COMMON/TEST/LDEBUG,LTEST
21 COMMON/CLRDC/LRDC(14,6)
22 IF(FN.GT.2.) GO TO 40
23 DV=0.
24 DO 10 N=1,3
25 10 DV=DV+D(N)*V(N,P,ME,N)
26 C!!! IS DIFFRACTION POSSIBLE?
27 IF(DV.LT.BCD(MP,ME,1).OR.DV.GT.BCD(MP,ME,2)) GO TO 39
28 C!!! COMPUTE RAY PATH
29 CALL RFOFPT(VH,XR,DOTP,DD,SHAG,VIC,XD,SP,VI,DV,ME,MP
30 2,LRDC(MP,"S"))
31 C!!! IS REFLECTION LEGAL?
32 IF(DOTP.LE.0.) GO TO 40
33 C!!! IS REFLECTION POINT OFF OF FINITE CYLINDER?
34 IF(XR(3).GT.ZC(1)+XR(1)*CTC(1).OR.
35 2XR(3).LT.ZC(2)+XR(1)*CTC(2)) GO TO 46
36 CNP=COS(FN*0.5*PI)
37 SNP=SIN(FN*0.5*PI)
38 DO 10 N=1,3
39 VECT=VP(NP,ME,N)*CNP+VN(MP,N)*SNP
40 10 XDP(N)=XD(N)+VECT*1.E-5
41 C!!! IS DIFFRACTED RAY SHADOWED BY A PLATE OR
42 C!!! A CYLINDER?
43 CALL PLAINTCADP,D,DHIT,MP,LHIT)
44 IF(LHIT) GO TO 40
45 CALL CYLINTXD,D,PHSR,DHIT,LHIT,.TRUE.)
46 IF(LHIT) GO TO 40
47 C!!! IS RAY BETWEEN REFLECTION AND DIFFRACTION SHADOWED?
48 CALL PLAINTCAN,VI,DHIT,MP,LHIT)
49 IF(LHIT.AND.(DHIT.LT.SP)) GO TO 40
50 C!!! IS RAY INCIDENT ON CYLINDER SHADOWED?
51 CALL PLAINTCAS,IC,DHIT,L,LHIT)
52 IF(LHIT.AND.(DHIT.LT.SRAG)) GO TO 40
53 Q1=0.
54 PP=0.
55 CD=0.
56 PL=0.
57 DO 20 N=1,3
58 C1=0.-VRINF,N)*V1(N)
59 PP=PP+VRINF,N)*V1(N)
60 CD=CD+VI(MP,N)*C1(N)
61 20 PP=PP+VRINF,IF,N)*C1(N)
62 PSOR=PT/2*PI,PP)
63 PSOR=PSOR/PSOR
64 IF(PSO.LT.0.) PSO=360.-PSO
65 PS=PIA(2)*E,PSO)
66 PS=PS*180

```

```

67 IF(PS.L1.0.) PS=360./PS
68 FNP=FN*180./1.E-4
69 IF(PSO.GT.FNP.OR.PS.GT.FNP) GO TO 40
70 SPHO=SIN(PSOK)
71 CPHO=COS(PSOK)
72 SPH=SIN(PSH)
73 CPH=COS(PSH)
74 C!!! COMPUTE POLARIZATION UNIT VECTORS FOR INCIDENT
75 C!!! AND DIFFRACTED FIELD ON PLATE
76 DO 30 N=1,3
77 PHO(N)=-VP(NP,NE,N)*SPHO+VN(NP,N)*CPHO
78 30 PH(N)=-VP(NP,NE,N)*SPH+VN(NP,N)*CPH
79 BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
80 BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
81 BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
82 BO(1)=PH(2)*D(3)-PH(3)*D(2)
83 BO(2)=PH(3)*D(1)-PH(1)*D(3)
84 BO(3)=PH(1)*D(2)-PH(2)*D(1)
85 THICK=BIAN2(SINH(VIC(1)*VIC(1)+VIC(2)*VIC(2)),VIC(3))
86 PHICK=BTAN2(VIC(2),VIC(1))
87 CALL SOURCE(EP,EG,EIX,EIY,EIZ,THICK,PHICK,VAS)
88 NG=DD*DD/DD**3
89 CALL NALD(UN,UB,VH)
90 CTIC=UN(1)*VI(1)+UN(2)*VI(2)
91 NH=BTAN2(-VIC(1)*UB(1)-VIC(2)*UB(2),-VIC(3))
92 SH=SIN(NH)
93 CH=COS(NH)
94 SST2=SH*SH+CH*CH*CTIC*CTIC
95 RH02=SHAG
96 RH01=SHAG*NG*CTIC/(RG*CTIC+2.*SHAG*SST2)
97 C!!! COMPUTE POLARIZATION UNIT VECTORS FOR INCIDENT
98 C!!! AND REFLECTED FIELDS AT CYLINDER
99 UIPHX=SIN(NH-.5*PI)*UB(1)
100 UIPHY=SIN(NH-.5*PI)*UB(2)
101 UIPHZ=COS(NH-.5*PI)
102 UIPPX=VIC(3)*UIPHY-VIC(2)*UIPRZ
103 UIPPY=VIC(1)*UIPRZ-VIC(3)*UIPRX
104 UIPPZ=VIC(2)*UIPRX-VIC(1)*UIPHY
105 UNPPX=VI(3)*UIPPY-VI(2)*UIPRZ
106 UNPPY=VI(1)*UIPRZ-VI(3)*UIPRX
107 UNPPZ=VI(2)*UIPRX-VI(1)*UIPRY
108 C!!! CALCULATE COMPONENTS OF FIELD INCIDENT ON CYLINDER
109 C!!! PERPENDICULAR AND PARALLEL TO PLATE OF INCIDENCE
110 EXPH=EXP(1J*PHICK*U..-TPI*ST/CI)/SHAG
111 EIPX=UIPHX*EIX+UIPHY*EIY+UIPRZ*EIZ
112 EIPY=UIPPX*EIX+UIPPY*EIY+UIPPZ*EIZ
113 C!!! COMPUTE COMPONENTS OF CYLINDER REFLECTED FIELD
114 ENPX=-SORT(RH01)*RH02*EXPH*EIPX
115 ENPY=-SORT(RH01)*RH02*EXPH*EIPY
116 ENX=ENPX*UIPHX+ENPY*UIPRY
117 ENY=ENPX*UIPHY+ENPY*UIPRZ
118 ENZ=ENPX*UIPRZ+ENPY*UIPRZ
119 C!!! COMPUTE COMPONENTS OF FIELD INCIDENT ON PLATE
120 C!!! EDGE PARALLEL AND PERPENDICULAR TO EDGE
121 EIPX=ENX*PHO(1)+EIPY*PHO(2)+ENZ*PHO(3)
122 EIPY=ENX*BOP(1)+EIPY*BOP(2)+ENZ*BOP(3)
123 EIPZ=ENX*BO(1)+EIPY*BO(2)+ENZ*BO(3)
124 EIPX=SQRT(1+VNP*(E,3)*D(2)-VNP*(E,2)*D(1)+2*VNP*(E,1)
125 2*D(3)-VNP*(E,3)*D(1)+2*VNP*(E,2)*D(1)-VNP*(E,1)*D(2))
126 C!!! COMPUTE REFLECTED-DIFFRACTED RAY CUSTIC
127 C!!! DISTANCES AND SPREAD FACTORS
128 UN1=UN(1)+UIPR1*UN(2)+UIPPY
129 UN2=UN(1)+UIPR1*UN(2)+UIPPY
130 UN3=UN1+UN2+2*UIPR1*UN(1)
131 UN4=UN1+UN2+2*UIPR1*UN(2)
132 UN5=UN1+UN2

```

```

133 UXH2X=UIPRX-2.*UIN2*UH(1)
134 UXH2Y=UIPHY-2.*UIN2*UH(2)
135 UXH2Z=UIPHZ
136 TH1=UIPPX*UP(1)+UIPPY*UG(2)
137 TH2=UIFPZ
138 TH21=UIPRX*UG(1)+UIPHY*UG(2)
139 TH22=UIPRZ
140 DET=TH1*TH22-TH12*TH21
141 QH11=1./SMAG*2.*CTHC*TH22*TP22/(RG*DET*DET)
142 QH12=-2.*CTHC*TH22*TH12/(RG*DET*DET)
143 QH22=1./SMAG*2.*CTHC*TH12*TH12/(RG*DET*DET)
144 QH1=QH11+QH12
145 QH2=QH12+QH22
146 QH=SQRT(QH1*QH1+QH2*QH2)
147 XH1X=(QH1*UXH1X-QH2*UXH2X)/QH
148 XH1Y=(QH1*UXH1Y-QH2*UXH2Y)/QH
149 XH1Z=(QH1*UXH1Z-QH2*UXH2Z)/QH
150 XH2X=-(VI(2)*XH1Z-VI(3)*XH1Y)
151 XH2Y=-(VI(3)*XH1X-VI(1)*XH1Z)
152 XH2Z=-(VI(1)*XH1Y-VI(2)*XH1X)
153 CXH1=V(MP,ME,1)*XH1X+V(MP,ME,2)*XH1Y+V(MP,ME,3)*XH1Z
154 CXH2=V(MP,ME,1)*XH2X+V(MP,ME,2)*XH2Y+V(MP,ME,3)*XH2Z
155 RH1E=RH1*RH1E+SP
156 RH1I=RH1I+SP
157 RH2E=RH2*RH2E+SP
158 RH2I=RH2I+SP
159 TPP=RH1I*RH2E-SBO*SBO/RH1E
160 GAN=XD(1)*D(1)+XD(2)*D(2)+XD(3)*D(3)
161 EXPH=EXP(COPLX(M,TPI*(GAN-SP)))/SQRT(RH1I*RH2I)
162 EXPI=EXP(COPLX(M,TPI*(GAN-SP)))/SQRT(RH1I*RH2I)
163 C!!! COMPUTE DIFFRACTION COEFFICIENTS
164 CALL LGRDS,DP,DPS,DPH,TPP,PS,PSO,SBO,FN,.FALSE.)
165 C!!! COMPUTE DIFFRACTED FIELDS
166 EDPH=-EIPH*DP*EXPI
167 EDPL=-EIPH*PS*EXPH
168 C!!! COMPUTE THETA AND PHI COMPONENTS OF DIFFRACTED
169 C!!! FIELD IN PCS
170 EDTH=EDPL*(FC(1)*DT(1)+SC(2)*DT(2)+SC(3)*DT(3))
171 2*EDPH*(PH(1)*DT(1)+PH(2)*DT(2)+PH(3)*DT(3))
172 EDPL=EDPL*(BC(1)*DP(1)+BC(2)*DP(2))
173 2*EDPH*(BC(1)*DP(1)+BC(2)*DP(2))
174 GO TO 901
175 LGRD(MP,ME)=.FALSE.
176 CONTINUE
177 EDTH=0.,0.
178 EDPL=0.,0.
179 C!!!
180 IF(.NOT.LTEST) RETURN
181 EDTH=0.,0.
182 EDPL=0.,0.
183 C!!! TESTING NCLDPL SUBROUTINE*)
184 EDTH=0.,0. EDPL=0.,0.
185 RETURN
186 END

```

RCLRPL

PURPOSE

To calculate the geometrical optics fields of a source ray which is reflected by the elliptic cylinder and then reflected by a given plate.

PERTINENT GEOMETRY

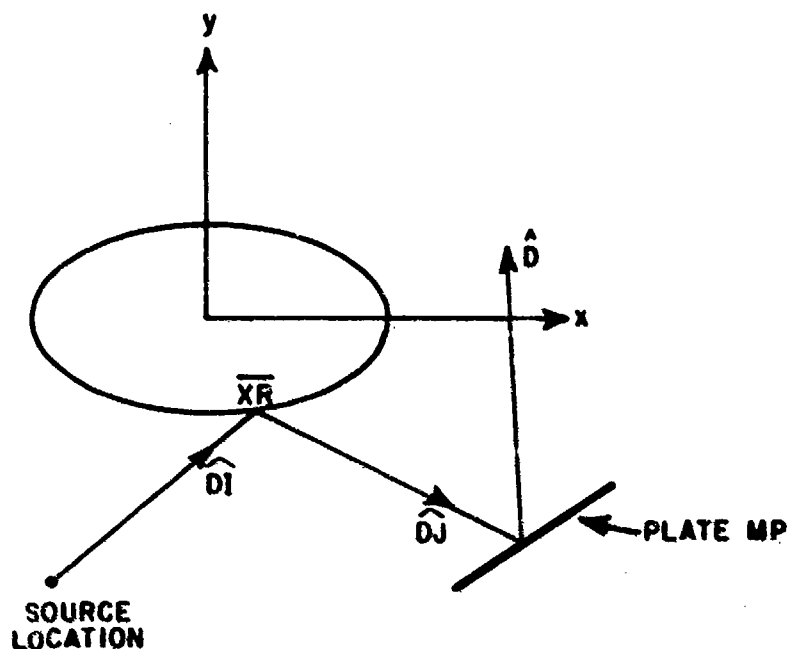


Figure 87--Illustration of ray reflected by cylinder and then reflected by a plate.

METHOD

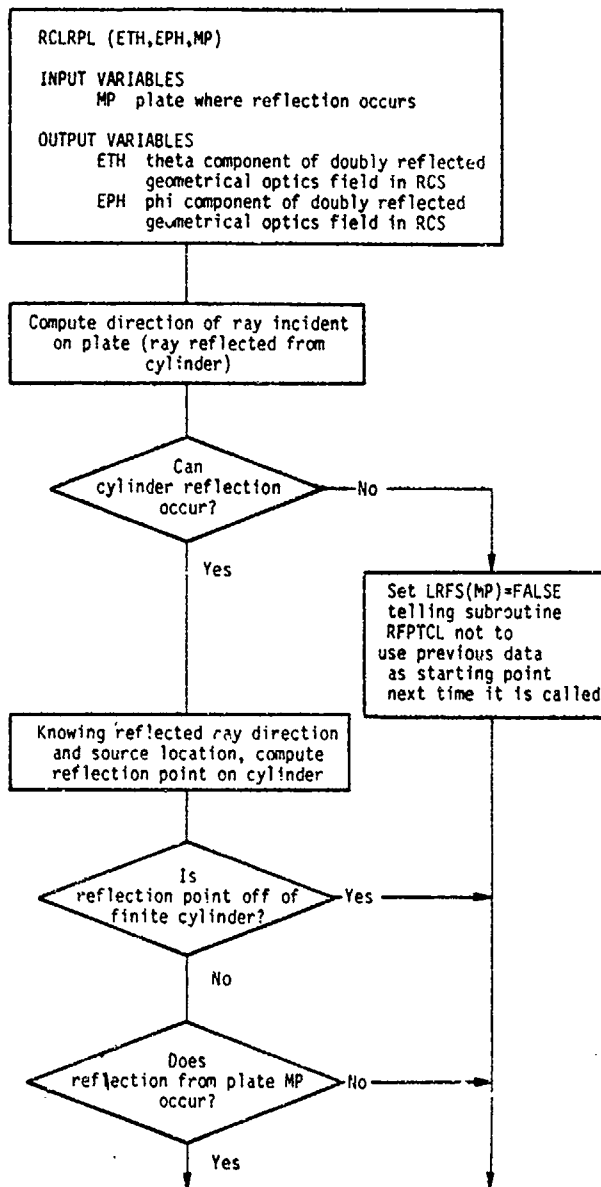
Subroutine RCLRPL functions as a service routine for subroutine SCLRPL, where the actual cylinder fields are computed. The geometrical optics reflected field components ETH and EPH computed in RCLRPL are used only for reference purposes (when LOUT is set true). The field components calculated in RCLRPL which are used in SCLRPL are the hard and soft components of the source field incident on the cylinder at the reflection point. These components, along with several other useful parameters are passed to subroutine SCLRPL through common block FUOGJ.

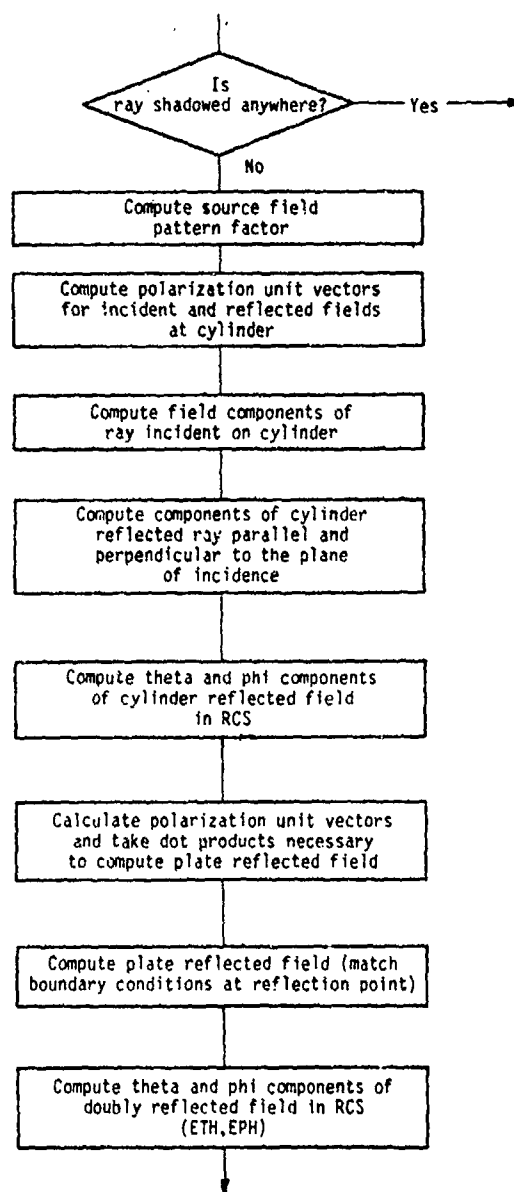
The geometrical optics fields determined in this subroutine, for the reflection from the cylinder, are calculated in the direction $\hat{D}$. This direction is found by imaging the observation direction into the plate, as illustrated in Figure 87. The cylinder reflected fields are found in a similar manner to those obtained in subroutine REFCYL. The plate reflected fields are found by satisfying the boundary conditions for the fields on the surface of the plate. The phase of the resultant double reflected field is referred to the reference coordinate system origin. The double reflected field thus has the form

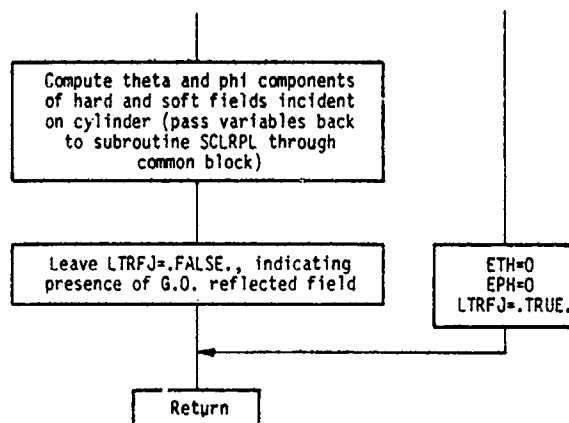
$$\vec{E}^{v,r} = W_m (E\hat{T}\hat{\theta} + E\hat{P}\hat{\phi}) \frac{e^{-jkR}}{R} ,$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM







SYMBOL DICTIONARY

A1 FIELD COMPONENTS OF RAY INCIDENT ON PLATE
 A2 NORMAL AND TANGENT TO THE PLATE
 A3 DETERMINANT OF POLARIZATION TRANSFORMATION
 C11
 C12 COEFFICIENTS USED TO CONVERT POLARIZATION FROM
 C21 THETA AND PHI COMPONENTS IN RCS TO COMPONENTS
 C22 NORMAL AND TANGENT TO PLATE (AND VICE-VERSA)
 D PROPAGATION DIRECTION AFTER PLATE REFL. IN (X,Y,Z)
 RCS COMPONENTS
 DD1 DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND
 CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 1 (2-D)
 DD2 DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND
 CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 2 (2-D)
 DHIT DISTANCE FROM SOURCE TO HIT POINT (FROM PLAIN)
 DH1 DISTANCE TO HIT POINT (FROM PLAIN AND CYLINT)
 DI X,Y, AND Z COMPONENTS OF INCIDENT RAY DIRECTION ON CYL IN RCS
 DJ X,Y,Z COMPONENTS OF PROPAGATION DIRECTION OF RAY
 INCIDENT ON PLATE
 EF PATTERN FACTOR OF THETA COMPONENT OF INCIDENT FIELD IN RCS
 (ALSO THETA COMPONENT OF CYL REFLECTED FIELD IN RCS)
 EG PATTERN FACTOR OF PHI COMPONENT OF INCIDENT FIELD
 IN RCS (ALSO PHI COMPONENT OF CYL REFL FIELD IN RCS)
 EHPH PHI COMPONENT OF HARD COMPONENT OF FIELD INCIDENT
 ON CYLINDER
 EHTH THETA COMPONENT OF THE HARD COMPONENT OF FIELD INC ON CYL
 (PARALLEL TO PLANE OF INCIDENCE)
 EIPP INCIDENT FIELD COMPONENT PARALLEL TO PLANE
 OF INCIDENCE ON CYLINDER
 EIPR INCIDENT FIELD COMPONENT PERPENDICULAR TO PLANE OF INC ON CYL
 ERPP COMPONENT OF CYLINDER REFLECTED FIELD PARALLEL
 TO PLANE OF INCIDENCE
 ERPR COMPONENT OF CYLINDER REFLECTED FIELD PERPENDICULAR
 TO PLANE OF INCIDENCE
 ERX }
 ERY } X,Y,Z COMPONENTS OF CYLINDER REFLECTED FIELD
 ERZ } IN RCS
 ESPH PHI COMPONENT OF SOFT COMPONENT OF FIELD INCIDENT
 ON CYL
 ES1H THETA COMPONENT OF SOFT COMPONENT OF FIELD INCIDENT
 ON CYL
 EX }
 EY } X,Y,Z COMPONENTS OF SOURCE FIELD PATTERN FACTOR
 EZ } IN RCS
 GAM PHASE CONSTANT
 LHIT SET TRUE IF RAY HITS PLATE (FROM PLAIN)
 LRFS SET TRUE IF REFL DATA IS AVAILABLE FROM PREVIOUS PATTERN
 ANGLE (OR FOR NEXT PATTERN ANGLE (WHEN LEAVING ROUTINE))
 LTRFJ SET TRUE IF G.O. REFLECTED-REFLECTED FIELDS
 DO NOT EXIST
 PH COMPLEX PHASE CONSTANT
 PHIR PHI COMPONENT OF INCIDENT RAY DIRECTION ON CYL IN RCS
 PHJR PHI COMPONENT OF RAY PROPAGATION DIRECTION
 BETWEEN CYLINDER AND PLATE IN RCS
 RH01 RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE
 AT REFLECTION POINT
 RH02 RAY SPREADING RADIUS IN PLANE NORMAL TO PLANE OF
 INCIDENCE AT REFLECTION POINT
 SMAG LENGTH OF RAY FROM REFL POINT ON CYL TO SOURCE
 SXN }
 SYN } X,Y,Z COMPONENTS OF UNIT VECTOR OF RAY FROM REFL.
 SZN } POINT ON CYLINDER TO SOURCE LOCATION IN RCS
 THIR THETA COMPONENT OF INCIDENT RAY DIRECTION ON CYLINDER
 THJR THETA COMPONENT OF RAY PROPAGATION DIRECTION
 BETWEEN CYLINDER AND PLATE

UIPPX } X,Y,Z COMPONENTS OF INCIDENT POLARIZATION UNIT VECTOR
 UIPPY }
 UIPPZ } PARALLEL TO PLANE OF INCIDENCE
 UIPRX } X,Y,Z COMPONENTS OF INC/REFL POLARIZATION UNIT VECTOR
 UIPRY }
 UIPHZ } PERPENDICULAR TO PLANE OF INCIDENCE
 URPPX } X,Y,Z COMPONENTS OF REFLECTED POLARIZATION UNIT VECTOR
 URPPY }
 URPPZ } PARALLEL TO PLANE OF INCIDENCE
 VT } X,Y,Z COMPONENTS OF POLARIZATION UNIT VECTOR TANGENT
 TO PLATE AND NORMAL TO RAY INCIDENT ON PLATE
 VXS } MATRIX DEFINING SOURCE COORDINATE SYS AXES IN RCS COMPONENTS
 XR } X,Y,Z COMPONENTS OF REFLECTION POINT LOCATION ON CYL
 XRS } REFLECTION POINT ON PLATE (ALSO CYL REFL. POINT IMAGE
 LOCATION IN PLATE) ALSO CYLINDER REFLECTION POINT

CODE LISTING

```

1 C-----
2 SUBROUTINE RCLRPL(ETH,EPH,MP)
3 C!!!
4 C!!! COMPUTES THE G.O. FIELD REFLECTED FROM THE ELLIPTIC CYLINDER
5 C!!! THEN REFLECTED FROM PLATE #MP
6 C!!!
7 DIMENSION UN(2),UB(2),DI(3),DJ(3),XRS(3),VT(3)
8 COMPLEX ETH,EPH,EX,EY,EZ,PH,EIPR,EIPP,ERX,ERY,ERZ,ERPR,ERPP
9 COMPLEX EF,EG,A1,A2,ESTH,ESPH,EHTH,EHPH,TRAN
10 LOGICAL LHIT,LRFS,LDEBUG,LTEST,LTRFJ
11 COMMON/FUDGJ/TRAN,ESTH,ESPH,EHTH,EHPH,XR(3),RG,R101,SMAG,LTRFJ
12 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
13 COMMON/SORINF/XS(3),VXS(3,3)
14 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VR(14,6,3),VN(14,3)
15 2,MEP(14),MPX
16 COMMON/PIS/PI,TPI,DPR,RPD
17 COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
18 COMMON/THPHUV/DT(3),DP(2)
19 COMMON/BNDSC/LDTS,VTS(2),BTS(4)
20 COMMON/TEST/LDEBUG,LTEST
21 COMMON/CLRFS/LRFS(14)
22 IF(LDEBUG) WRITE(6,900)
23 900 FORMAT(/' DEBUGGING RCLRPL SUBROUTINE')
24 LTRFJ=.FALSE.
25 IF(DTS.LT.-1.5) GO TO 12
26 C!!! COMPUTE DIRECTION OF RAY INCIDENT ON PLATE
27 CALL REFBP(PHJR,THJR,PHSR,THSR,MP)
28 SPHJ=SIN(PHJR)
29 CPHJ=COS(PHJR)
30 STHJ=SIN(THJR)
31 CTHJ=COS(THJR)
32 DJ(1)=CPHJ*STHJ
33 DJ(2)=SPHJ*STHJ
34 DJ(3)=CTHJ
35 DXY=XS(1)*CPHJ+XS(2)*SPHJ
36 IF(DXY.GT.0.) GO TO 10
37 DD1=BTS(1)*CPHJ+BTS(2)*SPHJ
38 DD2=BTS(3)*CPHJ+BTS(4)*SPHJ
39 C!!! CAN CYLINDER REFLECTION OCCUR?
40 IF(DD1.GT.DTS.AND.DD2.GT.DTS) GO TO 12
41 10 CONTINUE
42 C!!! COMPUTE CYLINDER REFLECTION POINT LOCATION
43 CALL REPTCL(PHJR,-MP,VR,DOTP,DD,S,LRFS(MP))
44 IF(LDEBUG) WRITE(6,*) VR,DOTP,DD,S,LRFS(MP)
45 IF(DOTP.LE.0.) GO TO 11
46 XR(1)=A*COS(VR)
47 XR(2)=B*SIN(VR)
48 XR(3)=XS(3)+S*CTHJ/STHJ
49 C!!! IS REFLECTION POINT ON CYLINDER?
50 IF(XR(3).GT.0.(1)+XR(1)*CTC(1).OR.
51 2XR(3).LT.ZC(2)+XR(1)*CTC(2)) GO TO 11
52 DO 15 N=1,3
53 15 XPS(N)=XR(N)
54 C!!! DOES REFLECTION FROM PLATE OCCUR?
55 CALL PLAINP(XRS,DJ,DHJT,-MP,LHIT)
56 IF(.NOT.LHIT) GO TO 11
57 C!!! IS RAY SHADOWED ANYWHERE?
58 CALL PLAINP(XRS,D,DHT,MP,LHIT)
59 IF(LHIT) GO TO 11
60 CALL CYLINT(XRS,D,PHSR,DHT,LHIT,.TRUE.)
61 IF(LHIT) GO TO 11
62 CALL PLAINP(XR,DJ,DHT,MP,LHIT)
63 IF(LHIT.AND.(DHT.LT.DHJT)) GO TO 11
64 SXN=XS(1)-XR(1)
65 SYN=XS(2)-XR(2)
66 SZN=-S*CTHJ/STHJ

```

```

67 SMAG=SQRT(SXN*SAH+SYN*SYN+SZN*SZN)
68 SXN=SXN/SMAG
69 SYN=SYN/SMAG
70 SZN=SZN/SMAG
71 PHIR=BTAN2(-SYN,-SXN)
72 THIR=BTAN2(SQRT(SXN*SXN+SYN*SYN),-SZN)
73 DI(1)=COS(PHIR)*SIN(THIR)
74 DI(2)=SIN(PHIR)*SIN(THIR)
75 DI(3)=COS(THIR)
76 CALL PLAIN(XS,DI,DHIT,0,LHIT)
77 IF(LHIT.AND.(DHIT.LT.SMAG)) GO TO 11
78 C!!! COMPUTE SOURCE PATTERN FACTOR
79 CALL SOURCE(EF,EG,EX,EY,EZ,THIR,PHIR,VXS)
80 IF(LDEBUG) WRITE(6,*) EF,EG
81 RG=DD*DD*DD/A/B
82 CALL NDNDB(UN,UB,VR)
83 CTHW=UN(1)*DJ(1)+UN(2)*DJ(2)
84 WR=BTAN2(SXN*UB(1)+SYN*UB(2),SZA)
85 SN=SIN(WR)
86 CW=COS(WR)
87 SST2=SW*SW+CW*CW*CTHW*CTHW
88 RHO2=SMAG
89 RHO1=SMAG*RG*CTHW/(RG*CTHW+2.*SMAG*SST2)
90 IF(LDEBUG) WRITE(6,*) RG,RHO1,RHO2,CTHI,SS.2
91 C!!! COMPUTE POLARIZATION UNIT VECTORS FOR
92 C!!! INCIDENT AND REFLECTED FIELD AT CYLINDER
93 UIPRX=SIN(WR-PI/2.)*UB(1)
94 UIPRY=SIN(WR-PI/2.)*UB(2)
95 UIPRZ=COS(WR-PI/2.)
96 IPPX=SYN*UIPRZ-SZN*UIPRY
97 UIPPY=SZN*UIPRX-SXN*UIPRZ
98 UIPPZ=SXN*UIPRY-SYN*UIPRX
99 URPPX=UIPRY*D(3)-UIPRZ*D(2)
100 URPPY=UIPRZ*D(1)-UIPRX*D(3)
101 URPPZ=UIPRX*D(2)-UIPRY*D(1)
102 PH=CEXP(CMPLX(0.,-TPI*SMAG))/SMAG
103 C!!! COMPUTE FIELD COMPONENTS OF RAY INCIDENT ON CYL.
104 EIPR=(UIPRX*EX+UIPRY*EY+UIPRZ*EZ)
105 EIPP=(UIPPX*EX+UIPPY*EY+UIPPZ*EZ)
106 C!!! COMPUTE LOCATION OF CYLINDER REFL. POINT
107 C!!! IMAGE IN PLATE MP
108 CALL IMAGE(XRS,XR,ANR,MP)
109 GAM=XRS(1)*D(1)+XRS(2)*D(2)+XRS(3)*D(3)
110 PH=PH+CEXP(CMPLX(0.,TPI*GAM))
111 SQRH=SQRT(RHO1*RHO2)
112 C!!! COMPUTE COMPONENTS OF CYLINDER REFL. FIELD
113 C!!! PARALLEL AND PERPENDICULAR TO PLANE OF INC
114 ERPR=-SQRH*PH*EIPR
115 ERPP=SQRH*PH*EIPP
116 TRAN=SQRH*PH
117 ERX=ERPR*UIPRX+ERPP*URPPX
118 ERY=ERPR*UIPRY+ERPP*URPPY
119 ERZ=ERPR*UIPRZ+ERPP*URPPZ
120 C!!! COMPUTE THETA AND PHI COMPONENTS OF CYLINDER
121 C!!! REFLECTED FIELD
122 EF=ERX*CPHJ*CTHJ+ERY*SPHJ*CTHJ-ERZ*STHJ
123 EG=-ERX*SPHJ+ERY*CPHJ
124 C!!! CALCULATE POLARIZATION VECTORS AND DOT PRODUCTS
125 C!!! NECESSARY TO COMPUTE PLATE REFLECTED FIELD
126 VT(1)=VN(MP,2)*D(3)-VN(MP,3)*D(2)
127 VT(2)=VN(MP,3)*D(1)-VN(MP,1)*D(3)
128 VT(3)=VN(MP,1)*D(2)-VN(MP,2)*D(1)
129 C11=VN(MP,1)*CPHJ*CTHJ+VN(MP,2)*SPHJ*CTHJ-VN(MP,3)*STHJ
130 C12=-VN(MP,1)*SPHJ+VN(MP,2)*CPHJ
131 C21=VT(1)*CPHJ*CTHJ+VT(2)*SPHJ*CTHJ-VT(3)*STHJ
132 C22=-VT(1)*SPHJ+VT(2)*CPHJ

```

```

133 C!!! COMPUTE FIELD REFLECTED FROM PLATE
134 A1=EF*C11+EG*C12
135 A2=EF*C21+EG*C22
136 C11=VN(MP,1)*DT(1)+VN(MP,2)*DT(2)+VN(MP,3)*DT(3)
137 C12=VN(MP,1)*DP(1)+VN(MP,2)*DP(2)
138 C21=VT(1)*DT(1)+VT(2)*DT(2)+VT(3)*DT(3)
139 C22=VT(1)*DP(1)+VT(2)*DP(2)
140 A3=C11*C22-C12*C21
141 C!!! COMPUTE THETA AND PHI REFLECTED FIELD COMPONENTS
142 ETH=(A1*C22+A2*C12)/A3
143 EPH=-(A2*C11+A1*C21)/A3
144 C!!! COMPUTE THETA AND PHI COMPONENTS OF HARD AND
145 C!!! SOFT COMPONENTS OF RAY INCIDENT ON CYLINDER
146 ERX=EIPH*UIPRX
147 ERY=EIPR*UIPRY
148 ERZ=EIPR*UIPRZ
149 ESTH=ERX*CPHJ*CTHJ+ERY*SPHJ*CTHJ-ERZ*STHJ
150 ESPH=-ERX*SPHJ+ERY*CPHJ
151 ERX=EIPP*URPPX
152 ERY=EIPP*URPPY
153 ERZ=EIPP*URPPZ
154 EHTH=ERX*CPHJ*CTHJ+ERY*SPHJ*CTHJ-ERZ*STHJ
155 EHPH=-ERX*SPHJ+ERY*CPHJ
156 GO TO 905
157 .12 LRFS(MP)=.FALSE.
158 .11 LTRFJ=.TRUE.
159 ETH=(0.,0.)
160 EPH=(0.,0.)
161 905 CONTINUE
162 IF(.NOT.LTEST) RETURN
163 WRITE(6,901)
164 901 FORMAT(/,' TESTING RCLRPL SUBROUTINE')
165 WRITE(6,*) ETH,EPH,MP
166 RETURN
167 END

```

REFBP

PURPOSE

To calculate incident ray direction needed in order to obtain reflected ray in a given direction off of a specified plate.

PERTINENT GEOMETRY

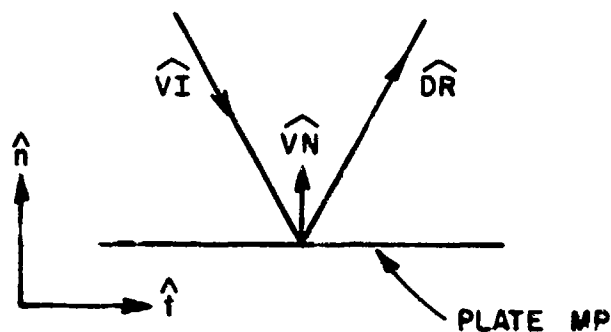


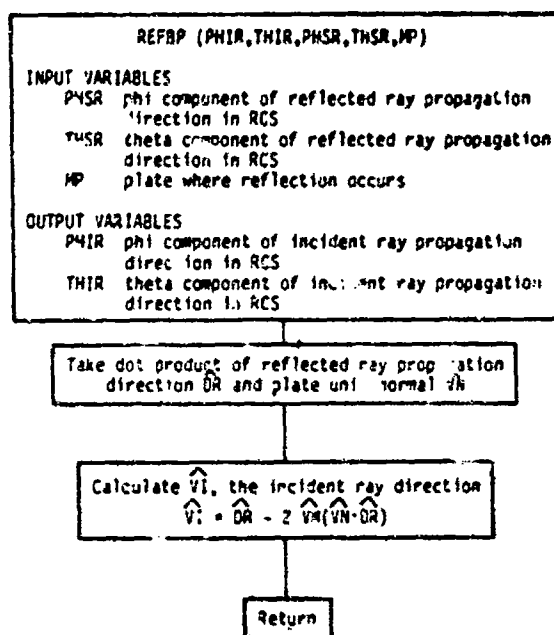
Figure 88--Illustration of incident and reflected rays on plate.

METHOD

$\hat{VI}$ is found by imaging $\hat{DR}$ into plate MP:

$$\hat{VI} = \hat{DR} - 2(\hat{VN} \cdot \hat{DR}) \hat{VN}.$$

FLOW DIAGRAM



SYMBOL DICTIONARY

CPS	COSINE OF PHSR
CTS	COSINE OF THSR
DN	CROSS PRODUCT OF DR AND VN
DR	REFLECTED RAY PROPAGATION DIRECTION IN X,Y,Z RCS COMPONENTS
END	ERROR DETECTION VARIABLE
MP	PLATE UPON WHICH REFLECTION OCCURS
PHIR	PHI COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN RCS
PHSR	PHI COMPONENT OF REFLECTED RAY PROPAGATION DIRECTION IN RCS
SPS	SINE OF PHSR
STS	SINE OF THSR
THIR	THETA COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN RCS
THSR	THETA COMPONENT OF REFLECTED RAY PROPAGATION DIRECTION IN RCS
V1	X,Y,Z COMPONENTS OF INCIDENT RAY PROPAGATION DIRECTION IN RCS
VN	CROSS PRODUCT OF PLATE NORMAL AND V1

CODE LISTING

```

1 C-----
2      SUBROUTINE REFBP(PHIR,THIR,PHSR,THSR,MP)
3 C!!!
4 C!!! DETERMINE INCIDENT RAY DIRECTION (PHIR,THIR)
5 C!!! IF RAY REFLECTED FROM PLATE #MP IS IN (PHSP,THSR) DIRECTION.
6 C!!!
7      DIMENSION DR(3),VI(3)
8      COMMON/CEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
9      2,MEP(14),MPX
10     COMMON/PIS/PI,TP1,DPR,RPD
11     CPS=COS(PHSR)
12     SPS=SIN(PHSR)
13     CTS=COS(THSR)
14     STS=SIN(THSR)
15     DR(1)=CPS*STS
16     DR(2)=SPS*STS
17     DR(3)=CTS
18 C!!! TAKE DOT PRODUCT OF DR AND VN
19     DN=VN(MP,1)*DR(1)+VN(MP,2)*DR(2)+VN(MP,3)*DR(3)
20 C!!! CALCULATE VI, THE INC RAY DIRECTION
21     DO 10 N=1,3
22 10    VI(N)=DR(N)-2.*DN*VN(MP,N)
23 C!!! CONVERT VI TO SPHERICAL ANGLES IN RCS
24     PHIR=ATAN2(VI(2),VI(1))
25     THIR=ATAN2(SQRT(VI(1)*VI(1)+VI(2)*VI(2)),VI(3))
26     VIN=VN(MP,1)*VI(1)+VN(MP,2)*VI(2)+VN(MP,3)*VI(3)
27     ERD=ABS(DN+VIN)
28     IF(ERD.GT.1.E-5) WRITE(6,1) ERD,PHSR,THSR
29 1    FORMAT(' ERROR IN REFBP= ',JF12.5)
30     RETURN
31     END

```

REFCAP

PURPOSE

To calculate the far-zone electric field resulting from the reflection of the source off of a given cylinder end cap.

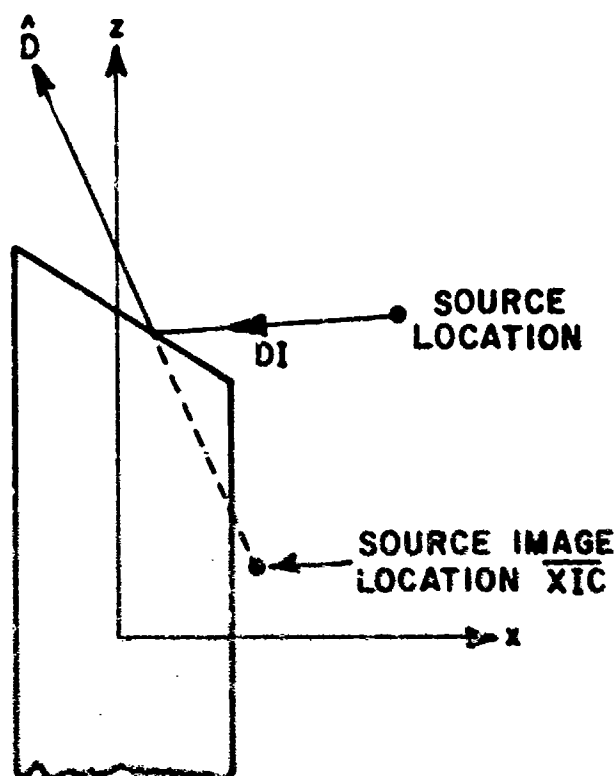


Figure 89--Illustration of source ray reflection from end cap.

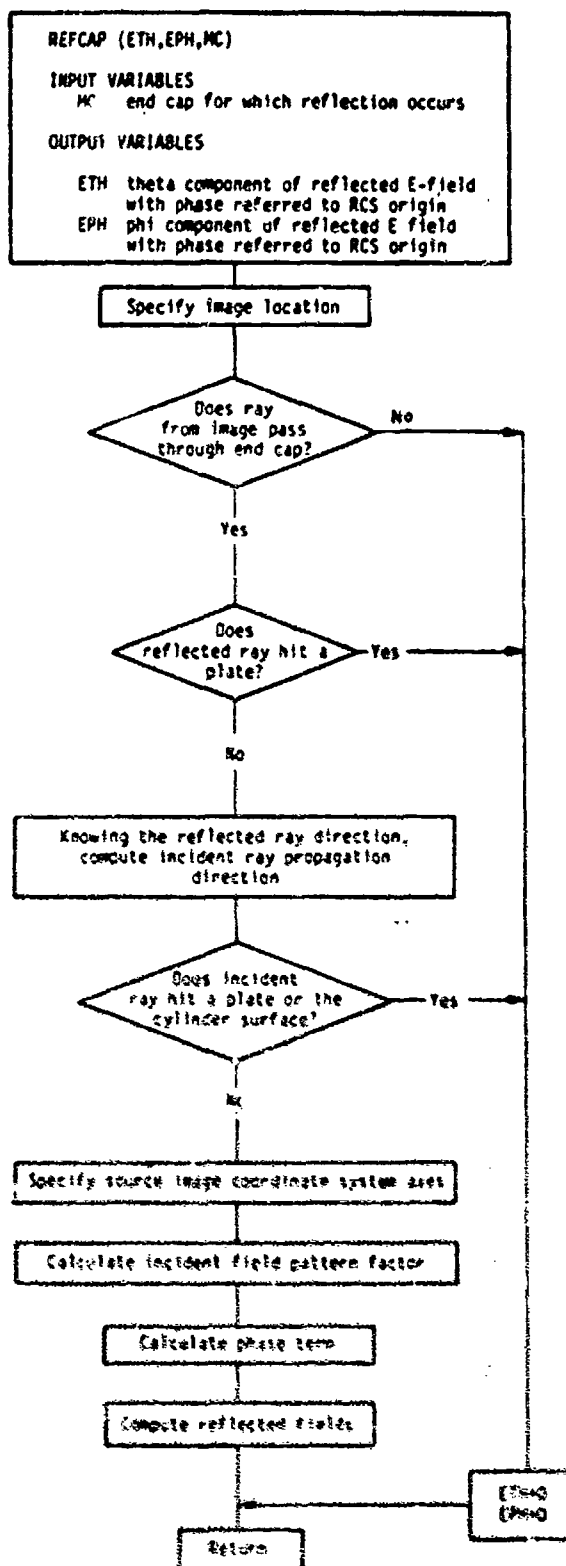
METHOD

The field reflected from a cylinder end cap is found using image theory. First the ray path is checked to insure that the reflection point is on the end cap and that the ray is not shadowed. The fields are then calculated using the SOURCE subroutine with the source coordinates oriented from image theory so that the proper boundary conditions are met at the surface of the end cap. The phase is referred to the reference coordinate system origin using the factor $e^{jkD \cdot XTC}$. The reflected field has the form

$$E^r(r, \theta, \phi) = W_m (E_{TH} \hat{\theta} + E_{PH} \hat{\phi}) \frac{e^{-jkR}}{R}$$

The factor $\frac{e^{-jkR}}{R}$ and source weight (W_m) are added elsewhere in

FLOW DIAGARM



SYMBOL DICTIONARY

DHT	DISTANCE FROM SOURCE TO HIT POINT ON END CAP (FROM CAPINT)
DHIT	DISTANCE FROM SOURCE TO HIT POINT ON PLATE (FROM PLAIN)
DI	UNIT VECTOR OF INCIDENT RAY PROPAGATION DIRECTION
DN	DOT PRODUCT OF REFLECTED RAY PROP DIR AND END CAP UNIT NORMAL
DNI	DOT PRODUCT OF INCIDENT RAY AND END CAP UNIT NORMAL
EF	PATTERN FACTOR FOR THETA COMPONENT OF INCIDENT E FIELD
EG	PATTERN FACTOR FOR PHI COMPONENT OF INCIDENT E FIELD
EPH	PHI COMPONENT OF REFLECTED E FIELD IN RCS
ETH	THETA COMPONENT OF REFLECTED E FIELD IN RCS
EX	PHASE TERM
GAM	PHASE TERM PARAMETER
LHIT	SET TRUE IF RAY HITS PLATE (FROM PLAIN)
MC	END CAP WHERE REFLECTION OCCURS
N	DO LOOP VARIABLE
NC	SIGN CHANGE VARIABLE
NI	DO LOOP VARIABLE
NJ	DO LOOP VARIABLE
VAX	X,Y,Z COMPONENTS DEFINING THE IMAGE SOURCE COORDINATE SYSTEM IN (XYZ) RCS COMPONENTS
VN	UNIT NORMAL TO END CAP IN RCS (X,Y,Z) COMPONENTS
XIS	SOURCE IMAGE LOCATION

CODE LISTING

```

1 C-----
2 SUBROUTINE REFCAP(ETH,EPH,MC)
3 C!!!
4 C!!! COMPUTES THE REFLECTED FIELD FROM THE END CAPS
5 C!!! OF THE ELLIPTIC CYLINDER
6 C!!!
7 COMPLEX ETH,EPH,EF,EG,EIX,EIY,EIZ,EX
8 DIMENSION XIS(3),DI(3),VN(3),VAX(3,3)
9 LOGICAL LHIT,LDEBUG,LTEST
10 COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
11 COMMON/SORINF/XS(3),VXS(3,3)
12 COMMON/INCINF/XIC(2,3),VXIC(3,3,2)
13 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
14 COMMON/PIS/PI,TPI,DPH,RPD
15 COMMON/TEST/LDEBUG,LTEST
16 IF(LDEBUG) WRITE(6,900)
17 900 FORMAT(/,' DEBUGGING REFCAP SUBROUTINE')
18 C!!! SPECIFY IMAGE LOCATION
19 DO 5 N=1,3
20 5 XIS(N)=XIC(MC,N)
21 C!!! DOES RAY FROM IMAGE PASS THRU DISK
22 CALL CAPINT(XIS,D,DHIT,MC,LHIT)
23 IF(.NOT.LHIT) GO TO 30
24 C!!! DOES REFL. RAY HIT A PLATE
25 CALL PLAINI(XIS,D,DHIT,0,LHIT)
26 IF(LHIT) GO TO 30
27 C!!! KNOWING OBS. DIR. COMPUTE THE INCIDENT RAY PROPAGATION
28 C!!! DIRECTION
29 NC=MC
30 IF(MC.GT.1) NC=-1
31 VN(1)=-NC*CNC(MC)
32 VN(2)=0.
33 VN(3)=NC*SNC(MC)
34 DN=VN(1)*D(1)+VN(2)*D(2)+VN(3)*D(3)
35 DO 10 N=1,3
36 10 DI(N)=D(N)-2.*DN*VN(N)
37 C!!! DOES RAY FROM SOURCE HIT A PLATE
38 CALL PLAINI(XS,DI,DHIT,0,LHIT)
39 IF(LHIT.AND.(DHIT.LT.DHIT)) GO TO 30
40 C!!! DOES RAY FROM SOURCE HIT THE CYLINDER
41 DNI=VN(1)*DI(1)+VN(2)*DI(2)+VN(3)*DI(3)
42 IF(DNI.GE.0.) GO TO 30
43 C!!! SPECIFY SOURCE IMAGE AXES
44 DO 20 NJ=1,3
45 DO 20 NI=1,3
46 20 VAX(NI,NJ)=VXIC(NI,NJ,MC)
47 C!!! CALCULATE INCIDENT FIELD PATTERN FACTOR
48 CALL SOURCE(EF,EG,EIX,EIY,EIZ,THSR,PHSR,VAX)
49 IF(LDEBUG) WRITE(6,*) XIS
50 IF(LDEBUG) WRITE(6,*) EF,EG
51 C!!! CALCULATE PHASE TERM (REFER PHASE TO RCS ORIGIN)
52 GAM=XIC(MC,1)*D(1)+XIC(MC,2)*D(2)+XIC(MC,3)*D(3)
53 EX=CEXP(CMPLX(0.,TPI*GAM))
54 ETH=EF*EX
55 EPH=EG*EX
56 RETURN
57 30 CONTINUE
58 ETH=(0.,0.)
59 EPH=(0.,0.)
60 IF(.NOT.LTEST) RETURN
61 WRITE(6,910)
62 910 FORMAT(/,' TESTING REFCAP SUBROUTINE')
63 WRITE(6,*) ETH,EPH,MC
64 RETURN
65 END

```

REFCYL

PURPOSE

To calculate the geometrical optics field due to reflection of the source field off of the cylinder surface and generate data used in subroutine SCTCYL.

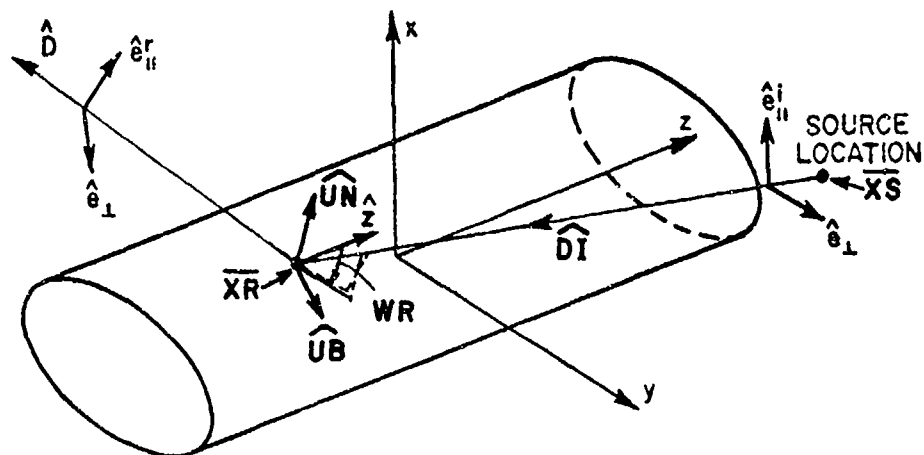


Figure 90 -- Geometry of ray reflected from cylinder.

$$\hat{e}_{\perp} = \text{UIPRX } \hat{x} + \text{UIPRY } \hat{y} + \text{UIPRZ } \hat{z}$$

$$\hat{e}_{\parallel}^i = \text{UIPPX } \hat{x} + \text{UIPPY } \hat{y} + \text{UIPPZ } \hat{z}$$

$$\hat{e}_{\parallel}^r = \text{URPPX } \hat{x} + \text{URPPY } \hat{y} + \text{URPPZ } \hat{z}$$

$$\hat{UN} = \text{UN}(1)\hat{x} + \text{UN}(2)\hat{y} = \text{normal to cylinder}$$

$$\hat{UB} = \text{UB}(1)\hat{x} + \text{UB}(2)\hat{y} = \text{tangent to cylinder}$$

$$\overline{XR} = \text{reflection point} = \hat{x} \text{XR}(1) + \hat{y} \text{XR}(2) + \hat{z} \text{XR}(3)$$

$$\overline{XS} = \hat{x} \text{XS}(1) + \hat{y} \text{XS}(2) + \hat{z} \text{XS}(3)$$

METHOD

Subroutine REFCYL functions as a service routine for subroutine SCTCYL, where the actual cylinder fields are computed. The geometrical optics reflected field components ETH and EPH computed in REFCYL are used only for reference purposes (when LOUT is set true). The field components calculated in REFCYL which are used in SCTCYL are the hard and soft components of the source field incident on the cylinder at the reflection point. These components, along with several other useful parameters are passed to subroutine SCTCYL through common block FUDG.

The geometrical optics fields [4] in the far field have the form

$$\bar{E}^r = \bar{E}^i(Q_R) \cdot \bar{R} \sqrt{\rho_1 \rho_2} \frac{e^{-jks}}{s}$$

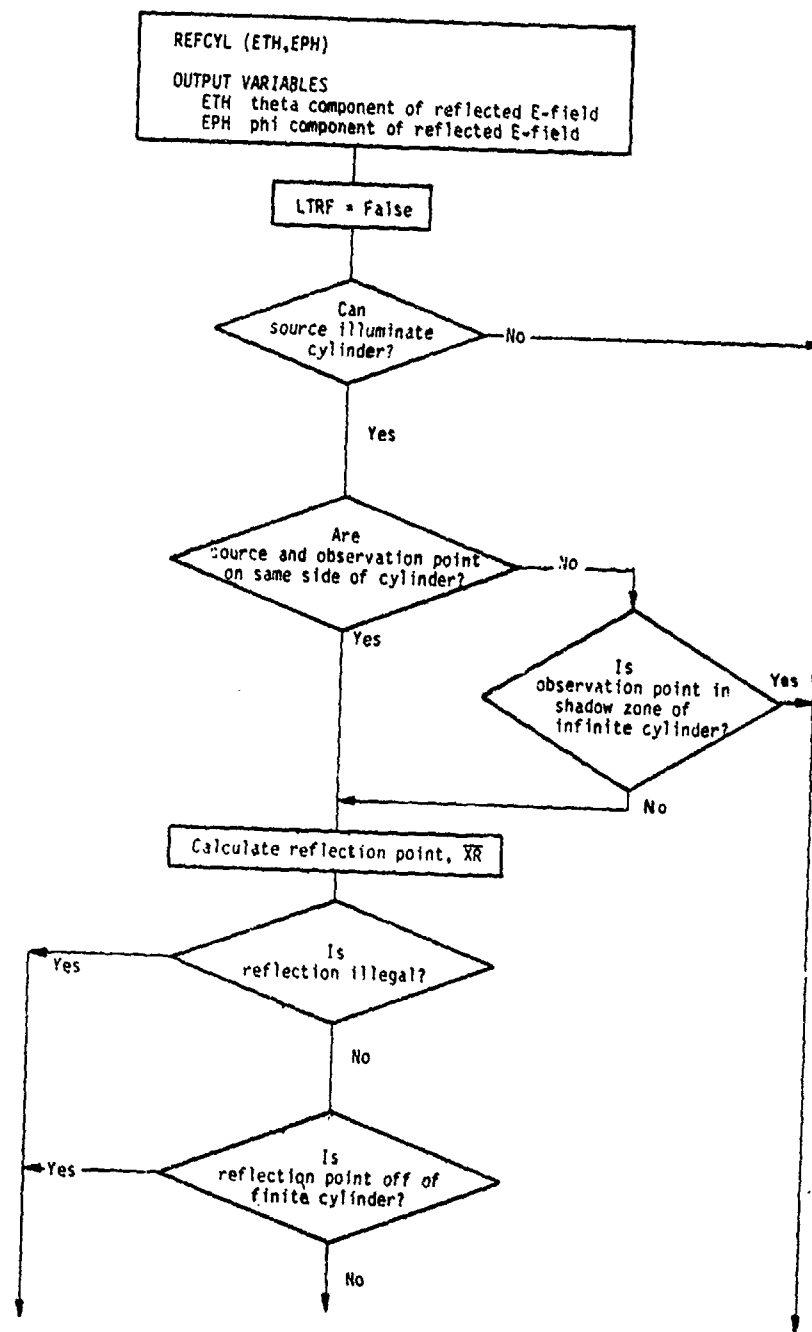
where $\bar{E}^i(Q_R)$ is the incident field at the reflection point, $\bar{R}$ is the dyadic reflection coefficient, s is the distance from the reflection point to the far field, and $\sqrt{\rho_1 \rho_2}/s$ is the far-field spread factor for the field. The caustic distances ρ_1 and ρ_2 and further details to the solution are given on pages 105-107 of Reference 1. The phase of the reflected field is referred to the reference co-

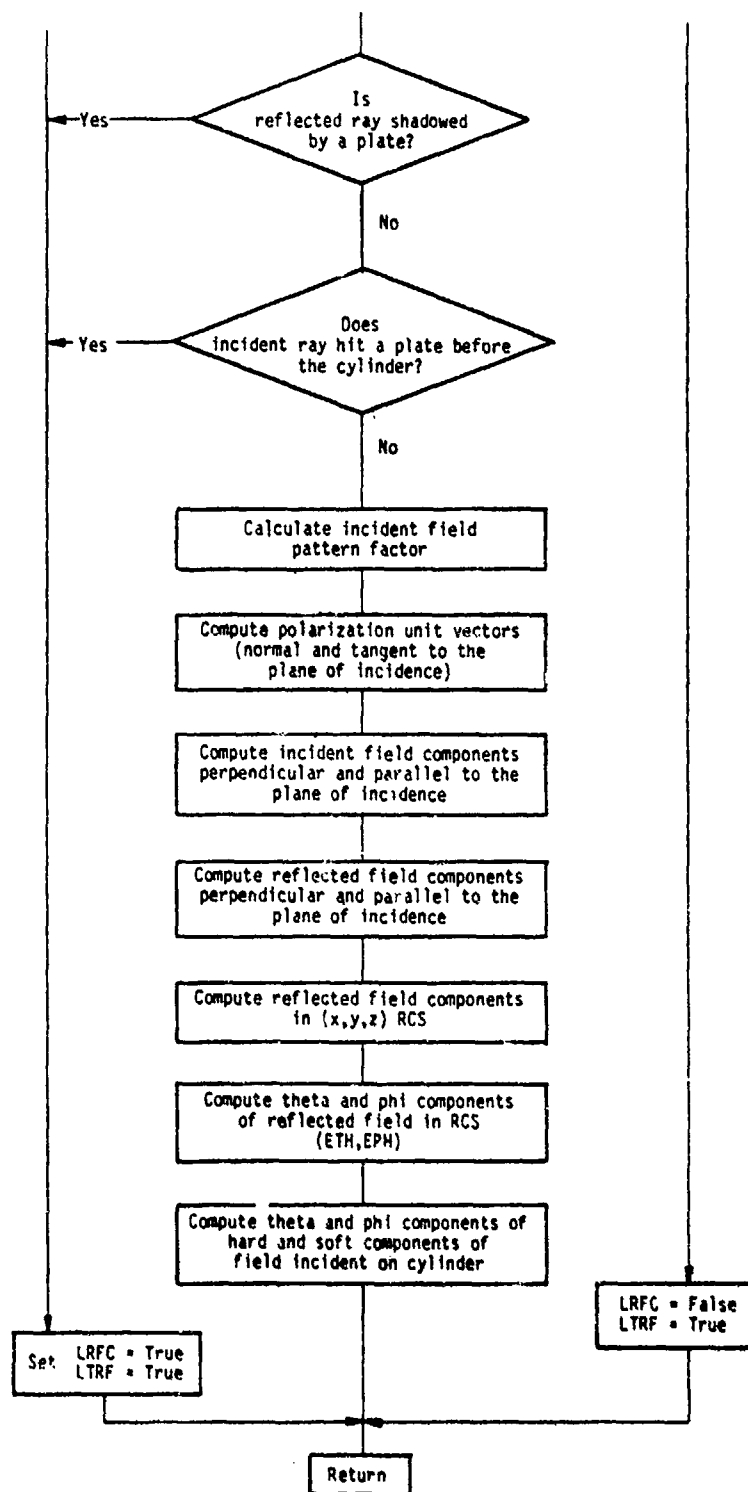
ordinate system origin so that $\frac{e^{-jks}}{s} = e^{jk\hat{D} \cdot \bar{X}R} \frac{e^{-jkR}}{R}$. The reflected field then has the form

$$\bar{E}^r = W_m (ETH\hat{\theta} + EPH\hat{\phi}) \frac{e^{-jkR}}{R}$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM





SYMBOL DICTIONARY

CTHI DOT PRODUCT OF CYLINDER NORMAL AND REFL PROP DIR UNIT VECTOR
 CW COSINE OF WR
 D PROPAGATION DIRECTION AFTER REFL. IN (X,Y,Z) RCS COMPONENTS
 DI2 DOT PRODUCT OF SOURCE VECTORS TANGENT TO CYLINDER. (2-D)
 DD NORMALIZATION CONSTANT FOR REFL. PT. UNIT NORMAL (FROM REPTCL)
 DD1 DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND
 CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 1 (2-D)
 DD2 DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND
 CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 2 (2-D)
 DHIT DISTANCE FROM SOURCE TO HIT POINT (FROM PLAIN)
 DI X,Y, AND Z COMPONENTS OF INCIDENT RAY DIRECTION IN RCS
 DOTP DIFFERENCE OF DOT PRODUCTS RETURNED FROM SUB REPTCL (2-D)
 DXY DOT PRODUCT OF VECTOR FROM ORIGIN TO SOURCE AND PROP. DIR (2-D)
 EF PATTERN FACTOR OF THETA COMPONENT OF INCIDENT FIELD IN RCS
 EG PATTERN FACTOR OF PHI COMPONENT OF INCIDENT FIELD IN RCS
 EHPH PHI COMPONENT OF THE HARD COMPONENT OF FIELD INC ON CYL
 ETHH THETA COMPONENT OF THE HARD COMPONENT OF FIELD INC ON CYL
 EIPP INCIDENT FIELD COMPONENT PARALLEL TO PLANE OF INCIDENCE
 EIPR INCIDENT FIELD COMPONENT PERPENDICULAR TO PLANE OF INC
 EPH PHI COMPONENT OF REFLECTED E-FIELD
 ERPP REFLECTED FIELD COMPONENT PARALLEL TO PLANE OF INCIDENCE
 ERPR REFLECTED FIELD COMPONENT PERPENDICULAR TO PLANE OF INC.
 ERX X,Y,Z COMPONENTS OF REFLECTED FIELD IN RCS
 ERY (ALSO USED TO DEFINE COMPONENTS INCIDENT ON
 ERZ CYLINDER)
 ESPH PHI COMPONENT OF THE SOFT COMPONENT OF FIELD INC ON CYL
 ESTH THETA COMPONENT OF THE SOFT COMPONENT OF FIELD INC ON CYL
 ETH THETA COMPONENT OF REFLECTED E FIELD
 EX }
 EY } PATTERN FACTOR OF X,Y,Z COMPONENTS OF INCIDENT FIELD IN RCS
 EZ }
 LHIT SET TRUE IF RAY HITS PLATE (FROM PLAIN)
 LRFC SET TRUE IF REFL DATA IS AVAILABLE FROM PREVIOUS PATTERN
 ANGLE (OR FOR NEXT PATTERN ANGLE WHEN LEAVING ROUTINE)
 LTRF SET TRUE IF G.O. REFLECTED FIELD DOES NOT EXIST
 PH PHASE AND MAGNITUDE CONSTANT FOR INCIDENT OR REFLECTED FIELD
 PHIR PHI COMPONENT OF INCIDENT RAY DIRECTION
 RG PARAMETER USED IN TRANSITION FUNCTION
 RH01 RAY SPREADING RADIUS IN PLANE OF CYL CURVATURE AT REFL. PT.
 RH02 RAY SPREADING RADIUS IN PLANE NORMAL TO PLANE
 OF INCIDENCE AT REFLECTION POINT
 S DISTANCE FROM SOURCE TO REFL. POINT IN X-Y PLANE
 SMAG DISTANCE FROM SOURCE TO REFLECTION POINT
 SQRH SPREADING FACTOR
 SW SINE OF WR
 SXN }
 SYN } X,Y, AND Z COMPONENTS OF UNIT VECTOR OF RAY FROM REFL.
 SZN } POINT TO SOURCE IN RCS
 THIR THETA COMPONENT OF INCIDENT RAY DIRECTION
 THAN PARAMETER USED IN TRANSITION FUNCTION
 TX1 X COMPONENT OF SOURCE VECTOR TANGENT TO TAN POINT 1 (2-D)
 TX2 X COMPONENT OF SOURCE VECTOR TANGENT TO TAN POINT 2 (2-D)
 TY1 Y COMPONENT OF SOURCE VECTOR TANGENT TO TAN POINT 1 (2-D)
 TY2 Y COMPONENT OF SOURCE VECTOR TANGENT TO TAN POINT 2 (2-D)
 UB X,Y COMPONENTS OF UNIT VECTOR TANGENT TO CYLINDER
 REFLECTION POINT IN RCS (2-D)
 UIPPX }
 UIPPY } X,Y,Z COMPONENTS OF INCIDENT FIELD POLARIZATION UNIT VECTOR
 UIPPZ } PARALLEL TO PLANE OF INCIDENCE
 UIPRX }
 UIPRY } X,Y,Z COMPONENTS OF INC/REFL FIELD POLARIZATION UNIT VECTOR
 UIPRZ } PERPENDICULAR TO PLANE OF INCIDENCE
 UN X,Y COMPONENTS OF UNIT NORMAL TO CYLINDER REFL
 POINT IN RCS (2-D)

URPPA	}	X,Y,Z COMPONENTS OF REFL FIELD POLARIZATION UNIT VECTOR
URPPY		
URPPZ		
VR		PARALLEL TO PLANE OF INCIDENCE
		ELL. ANGLE DEFINING REFLECTION POINT IN ERCS.
VXS		X,Y,Z COMPONENTS OF UNIT VECTORS DEFINING SOURCE
		COORDINATE SYSTEM AXES IN RCS
WR		PHI ANGLE DEFINING PROPAGATION
		DIRECTION IN CYL REFL. POINT COORD SYSTEM
XR		LOCATION OF REFLECTION POINT IN (X,Y,Z) REF COORD SYS.

CODE LISTING

```

1 C-----
2 SUBROUTINE REFCYL(ETH,EPH)
3 C!!!
4 C!!! COMPUTES THE REFLECTED FIELD OF THE ELLIPTIC CYLINDER
5 C!!!
6 DIMENSION UN(2),UB(2),DI(3)
7 COMPLEX ETH,EPH,EX,EY,EZ,PH,EIPR,EIPP,ERX,ERY,ERZ,ERPR,ERPP
8 COMPLEX ESTH,ESPH,EHTH,EHPH,TRAN,EF,EG
9 LOGICAL LHIT,LHFC,LTRF,LDEBUG,LTEST
10 COMMON/FUDG/TRAN,ESTH,ESPH,EHTH,EHPH,XR(3),RG,RHOI,SMAG,LTRF
11 COMMON/GEOMEL/A,R,ZC(2),SNC(2),CNC(2),CTC(2)
12 COMMON/SORINF/XS(3),VXS(3,3)
13 COMMON/PIS/PI,TPI,DPR,RPD
14 COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
15 COMMON/THPHIV/DT(3),DP(2)
16 COMMON/ENDSCL/DTS,VTS(2),BTS(4)
17 COMMON/TEST/LDEBUG,LTEST
18 COMMON/CLRFC/LRFC
19 IF(LDEBUG) WRITE(6,900)
20 900 FORMAT(/,' DEBUGGING REFCYL SUBROUTINE')
21 LTRF=.FALSE.
22 C!!! CAN SOURCE ILLUMINATE CYLINDER?
23 IF(DTS.LT.-1.5) GO TO 12
24 DXY=XS(1)*CPS+XS(2)*SPS
25 C!!! IS SOURCE AND OBSERVATION POINT ON SAME SIDE OF CYLINDER?
26 IF(DXY.GT.0.) GO TO 10
27 DI2=DTS
28 TX1=BTS(1)
29 TY1=BTS(2)
30 TX2=BTS(3)
31 TY2=BTS(4)
32 DD1=TX1*CPS+TY1*SPS
33 DD2=TX2*CPS+TY2*SPS
34 C!!! IS OBSERVATION POINT IN SHADOW ZONE OF INFINITE CYLINDER?
35 IF(DD1.GT.DI2.AND.DD2.GT.DI2) GO TO 12
36 10 CONTINUE
37 C!!! CALCULATE REFLECTION POINT
38 CALL RFPCTL(PHSR,0,VR,DOTP,DD,S,LRFC)
39 IF(LDEBUG) WRITE(6,*) VR,DOTP,DD,S,LRFC
40 C!!! IS REFLECTION ILLEGAL?
41 IF(DOTP.LE.0.) GO TO 11
42 XR(1)=A*COS(VR)
43 XR(2)=B*SIN(VR)
44 XR(3)=XS(3)+S*CTHS/STHS
45 C!!! IS REFLECTION POINT OFF OF FINITE CYLINDER?
46 IF(XR(3).GT.ZC(1)+XR(1)*CTC(1).OR.
47 2XR(3).LT.ZC(2)+XR(1)*CTC(2)) GO TO 11
48 C!!! IS REFLECTED RAY SHADOWED BY A PLATE?
49 CALL PLAIN(XR,D,LHIT,0,LHIT)
50 IF(LHIT) GO TO 11
51 SXN=XS(1)-XR(1)
52 SYN=XS(2)-XR(2)
53 SZN=S*CTHS/STHS
54 SMAG=SQRT(SXN*SXN+SYN*SYN+SZN*SZN)
55 SXN=SXN/SMAG
56 SYN=SYN/SMAG
57 SZN=SZN/SMAG
58 PHIR=ATAN2(-SYN,-SXN)
59 THIR=ATAN2(SQRT(SXN*SXN+SYN*SYN),-SZN)
60 RI(1)=CLS(PHIR)*SIN(THIR)
61 RI(2)=SIN(PHIR)*SIN(THIR)
62 RI(3)=CLS(THIR)
63 C!!! DOES INCIDENT RAY HIT PLATE BEFORE CYLINDER?
64 CALL PLAIN(XS,DI,LHIT,0,LHIT)
65 IF(LHIT.AND.(LHIT.LT.SMAG)) GO TO 11
66 C!!! CALCULATE INCIDENT FIELD PATTERN FACTORS

```

```

01 CALL SOURCE(EF,EG,EX,EY,EZ,THIR,PHIR,VX5)
02 IF(LDEBUG) WRITE(6,*) EF,EG
03 HG=DD*DD/DD/A/B
04 CALL NANDB(UN,UB,VR)
05 CTHI=UN(1)*D(1)+UN(2)*D(2)
06 WH=BTAN2(SXN*UB(1)+SYN*UB(2),SZN)
07 SW=SIN(WH)
08 CW=COS(WH)
09 SST2=SW*SW+CW*CW*CTHI*CTHI
10 RHO2=SMAG
11 RHO1=SMAG*HG*CTHI/(RG*CTHI+2.*SMAG*SST2)
12 IF(LDEBUG) WRITE(6,*) RG,RHO1,RHO2,CTHI,SST2
13 C!!! COMPUTE FIELD POLARIZATION UNIT VECTORS (PERPENDICULAR
14 C!!! AND PARALLEL TO PLANE OF INCIDENCE)
15 UIPRX=SIN(WH-PI/2.)*UB(1)
16 UIPHY=SIN(WH-PI/2.)*UB(2)
17 UIPHZ=COS(WH-PI/2.)
18 UIPPX=SYN*UIPHZ-SZN*UIPHY
19 UIPPY=SZN*UIPRX-SXN*UIPRZ
20 UIPPZ=SXN*UIPHY-SYN*UIPRX
21 UHPPX=UIPHY*D(3)-UIPRZ*D(2)
22 URPPY=UIPHZ*D(1)-UIPRX*D(3)
23 URPPZ=UIPRX*D(2)-UIPHY*D(1)
24 PH=CEXP(CMPLX(0.,-PI)*SMAG)/SMAG
25 C!!! COMPUTE INCIDENT FIELD COMPONENTS PERPENDICULAR AND
26 C!!! PARALLEL TO PLANE OF INCIDENCE
27 EIPR=(UIPRX*EX+UIPHY*EY+UIPPZ*EZ)
28 EIPP=(UIPPX*EX+UIPPY*EY+UIPPZ*EZ)
29 PH=PH*CEXP(CMPLX(0.,PI)*(XR(1)*D(1)+XR(2)*D(2)+XR(3)*D(3)))
30 SORH=SQRT(RHO1*RHO2)
31 C!!! COMPUTE REFLECTED FIELD COMPONENTS PERPENDICULAR AND
32 C!!! PARALLEL TO PLANE OF INCIDENCE
33 ERPH=-SORH*PH*EIPR
34 ERPP=SORH*PH*EIPP
35 TRAN=SORH*PH
36 C!!! COMPUTE REFLECTED FIELD COMPONENTS IN (XYZ) RCS COMPONENTS
37 ERX=ERPH*UIPRX+ERPP*URPPX
38 ERY=ERPH*UIPHY+ERPP*URPPY
39 ERZ=ERPH*UIPHZ+ERPP*URPPZ
40 C!!! COMPUTE THETA AND PHI COMPONENTS OF REFLECTED FIELD IN RCS
41 ETH=ERX*DT(1)+ERY*DT(2)+ERZ*DT(3)
42 EPH=ERX*DP(1)+ERY*DP(2)
43 C!!! COMPUTE THETA AND PHI COMPONENTS OF HARD AND SOFT
44 C!!! COMPONENTS OF FIELD INCIDENT ON CYLINDER
45 ENX=EIPR*UIPRX
46 ENY=EIPR*UIPHY
47 ENZ=EIPR*UIPHZ
48 ESTH=ENX*DT(1)+ENY*DT(2)+ENZ*DT(3)
49 ESPH=ENX*DP(1)+ENY*DP(2)
50 ENX=EIPP*UIPRX
51 ENY=EIPP*UIPHY
52 ENZ=EIPP*UIPHZ
53 ESTH=ENX*DT(1)+ENY*DT(2)+ENZ*DT(3)
54 ENPH=ENX*DP(1)+ENY*DP(2)
55 GO TO V05
56 C2 LKPC=.FALSE.
57 C1 LKPC=.TRUE.
58 ETH=(0.,0.)
59 EPH=(0.,0.)
60 W05 CONTINUE
61 IF(.NOT.LTEST) RETURN
62 WRITE(6,V10)
63 V10 FORMAT(1,' TESTING REFCYL SUBROUTINE')
64 WRITE(6,*) ETH,EPH
65 RETURN
66 END

```

REFPLA

PURPOSE

To calculate the far-zone electric field due to single reflection off of a given plate.

PERTINENT GEOMETRY

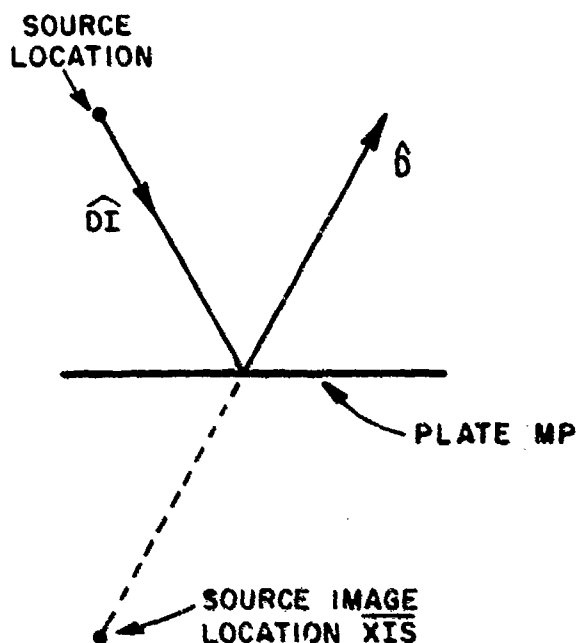


Figure 91-- Geometry for source ray reflection from plate

$$\vec{XIS} = \hat{x} XIS(1) + \hat{y} XIS(2) + \hat{z} XIS(3)$$

$$\hat{DI} = \hat{x} DI(1) + \hat{y} DI(2) + \hat{z} DI(3)$$

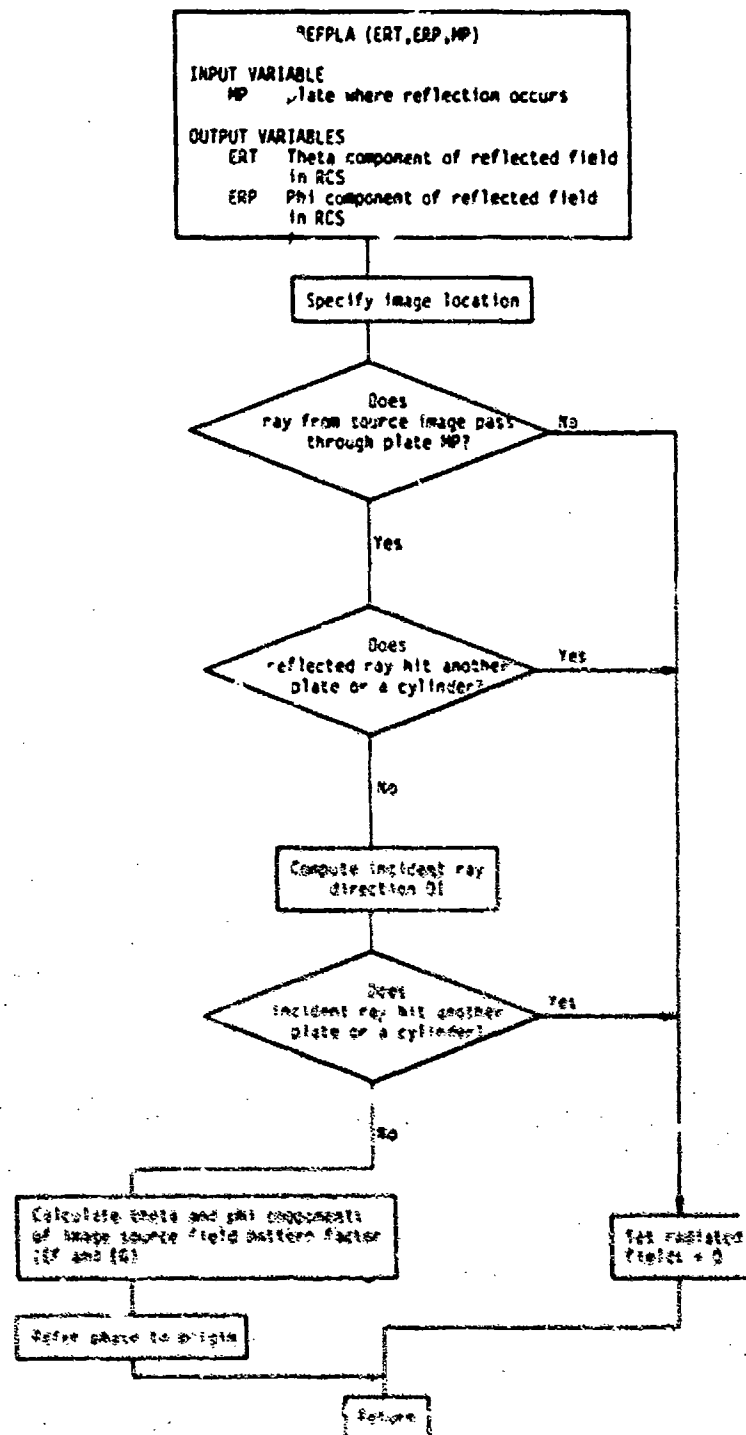
METHOD

The reflected field from a plate is found using image theory. First the ray path is checked to insure that the reflection point is on the plate and that the ray path is not shadowed. The fields are then calculated using the SOURCE subroutine with the source coordinates oriented from image theory so that the proper boundary conditions are met at the surface of the plate. The phase is referred to the reference coordinate system origin using the factor $e^{jk \cdot \vec{XIS}}$. The reflected field has the form

$$\vec{E}^r(r, \theta, \phi) = W_m (E R \hat{\theta} + E R P \hat{\phi}) \frac{e^{-jkR}}{R} .$$

The factor $\frac{e^{-jkR}}{R}$ and the source current weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM



SYMBOL DICTIONARY

CPHI COSINE OF PHIR
 CPHS COSINE OF PHSR
 CTHI COSINE OF THIR
 CTHS COSINE OF THSR
 D X,Y,Z COMPONENTS OF RAY PROPAGATION DIRECTION
 AFTER REFLECTION IN RCS
 DRIT DISTANCE FROM SOURCE TO REFLECTION POINT
 (FROM PLAIN)
 DHT DISTANCE FROM SOURCE TO HIT POINT (FROM PLAIN
 AND CYLIND)
 DI X,Y,Z COMPONENTS OF INCIDENT RAY PROPAGATION
 DIRECTION IN RCS
 EF PATTERN FACTOR FOR THETA COMPONENT OF SOURCE
 FIELD IN RCS
 EG PATTERN FACTOR FOR PHI COMPONENT OF SOURCE
 FIELD IN RCS
 EIX NOT USED
 Eiy NOT USED
 EIZ NOT USED
 EX COMPLEX PHASE FACTOR (CEXP(J*TPI*GAM))
 GAM PHASE DISTANCE TO ORIGIN (DOT PRODUCT OF IMAGE
 LOCATION AND REFLECTED RAY PROPAGATION DIRECTION)
 LHIT SET TRUE IF RAY INTERSECTS A PLATE OR CYLINDER
 (FROM PLAIN OR CYLIND)
 MP PLATE FROM WHICH REFLECTION OCCURS
 N DO LOOP VARIABLE
 NI DO LOOP VARIABLE
 NJ DO LOOP VARIABLE
 PHIR PHI COMPONENT OF INCIDENT RAY PROPAGATION
 DIRECTION IN RCS
 PHSR PHI COMPONENT OF RAY PROPAGATION DIRECTION
 AFTER REFLECTION IN RCS
 SPHI SINE OF PHIR
 SPHS SINE OF PHSR
 STHI SINE OF THIR
 THIR THETA COMPONENT OF INCIDENT RAY PROPAGATION
 DIRECTION IN RCS
 THSR THETA COMPONENT OF RAY PROPAGATION DIRECTION
 AFTER REFLECTION IN RCS
 VAX X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF THE
 SOURCE IMAGE COORDINATE SYSTEM AXES IN RCS
 XI TRIPLY DIMENSIONED ARRAY OF IMAGE LOCATIONS
 IIS X,Y,Z COMPONENTS OF SOURCE IMAGE LOCATION
 (SINGLE REFLECTION FROM PLATE MP)
 XS SOURCE LOCATION IN (X,Y,Z) REF COORD SYS.

```

1 C-----
2 SUBROUTINE REFPLA(ERT,ERP,MP)
3 C!!!
4 C!!! DETERMINES THE REFLECTED FIELD FROM PLATE #MP WITH PHASE
5 C!!! REFERRED TO THE ORIGIN.
6 C!!!
7 COMPLEX EF,EG,EX,ERT,ERP,EIX,EIY,EIZ
8 DIMENSION XIS(3),DI(3),VAX(3,3)
9 LOGICAL LHIT
10 LOGICAL LDEBUG,LTEST
11 COMMON/TEST/LDEBUG,LTEST
12 COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
13 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
14 2,MEP(14),MPX
15 COMMON/SORINF/XS(3),VXS(3,3)
16 COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
17 COMMON/PI5/PI,TPI,DPR,RPD
18 IF (LDEBUG) WRITE (6,101)
19 101 FORMAT (/,' DEBUGGING REFPLA SUBROUTINE')
20 C!!! SPECIFY IMAGE LOCATION.
21 DO 5 N=1,3
22 5 XIS(N)=XI(MP,MP,N)
23 C!!! DOES RAY FROM SOURCE IMAGE PASS THRU PLATE
24 CALL PLAINI(XIS,D,DHIT,-MP,LHIT)
25 IF(.NOT.LHIT) GO TO 30
26 C!!! DOES REFL. RAY HIT ANOTHER PLATE.
27 CALL PLAINI(XIS,D,DHT,MP,LHIT)
28 IF(LHIT) GO TO 30
29 C!!! DOES REFL. RAY HIT A CYLINDER.
30 CALL CYLINT(XIS,D,PHSR,DHT,LHIT,.TRUE.)
31 IF(LHIT) GO TO 30
32 C!!! KNOWING RAD. DIR. COMPUTE THE INCIDENT RAY DIRECTION
33 CALL REFBP(PHIR,THIR,PHSR,THSR,MP)
34 IF (LDEBUG) WRITE (6,*) PHIR,THIR,PHSR,THSR,MP
35 SPHI=SIN(PHIR)
36 CPHI=COS(PHIR)
37 STHI=SIN(THIR)
38 CTHI=COS(THIR)
39 DI(1)=CPHI*STHI
40 DI(2)=SPHI*STHI
41 DI(3)=CTHI
42 C!!! DOES RAY FROM SOURCE HIT ANOTHER PLATE.
43 CALL PLAINI(XS,DI,DHT,MP,LHIT)
44 IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 30
45 C!!! DOES RAY FROM SOURCE HIT A CYLINDER.
46 CALL CYLINT(XS,DI,PHIR,DHT,LHIT,.FALSE.)
47 IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 30
48 DO 20 NJ=1,3
49 DO 20 NI=1,3
50 20 VAX(NI,NJ)=VXI(NI,NJ,MP)
51 C!!! CALCULATE SOURCE FIELD PATTERN FACTOR
52 CALL SOURCE(EF,EG,EIX,EIY,EIZ,THSR,PHSR,VAX)
53 IF (LDEBUG) WRITE (6,*) EF,EG
54 C!!! COMPUTE PHASE REFERRED TO THE ORIGIN.
55 GAM=XI(MP,MP,1)*D(1)+XI(MP,MP,2)*D(2)+XI(MP,MP,3)*D(3)
56 EX=CEXP(CMPLX(0.,TPI*GAM))
57 ERT=EF*EX
58 ERP=EG*EX
59 GO TO 1
60 30 CONTINUE
61 ERT=(0.,0.)
62 ERP=(0.,0.)
63 1 IF (.NOT.LTEST) GO TO 2
64 WRITE (6,3)
65 3 FORMAT (/,' TESTING REFPLA SUBROUTINE')
66 WRITE (6,*) ERT,ERP,MP
67 2 RETURN
68 END

```

RFDFIN

PURPOSE

To determine the reflection point on an elliptic cylinder for a given source and observation location in the near field of the cylinder.

PERTINENT GEOMETRY

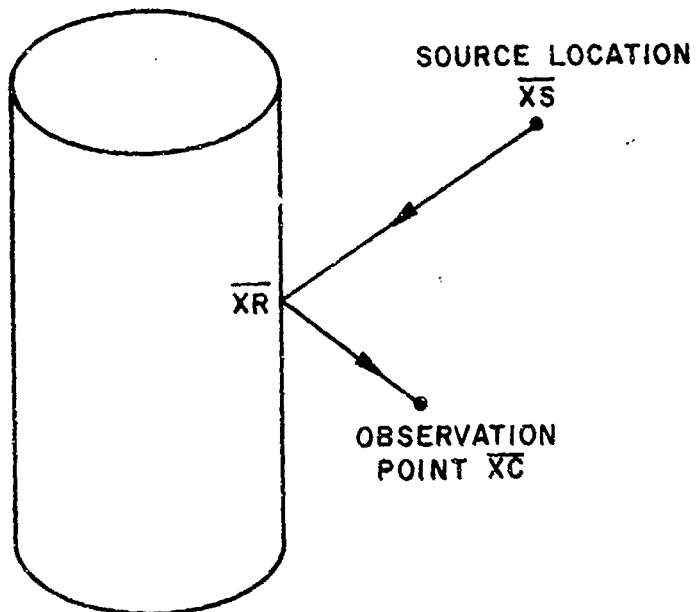


Figure 92-- Illustration of a reflection point on a cylinder for a near field observation point.

$$\overline{XS} = \hat{x} \text{ XS}(1) + \hat{y} \text{ XS}(2) + \hat{z} \text{ XS}(3)$$

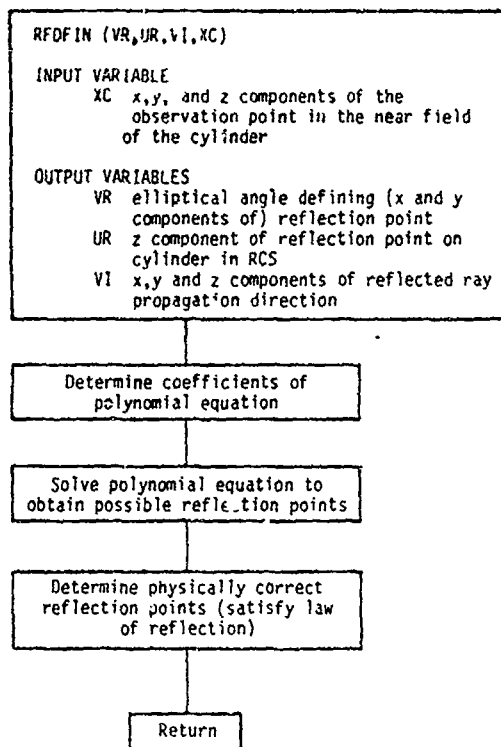
$$\overline{XR} = \hat{x} \text{ XR}(1) + \hat{y} \text{ XR}(2) + \hat{z} \text{ XR}(3)$$

$$\overline{XC} = \hat{x} \text{ XC}(1) + \hat{y} \text{ XC}(2) + \hat{z} \text{ XC}(3)$$

METHOD

This subroutine solves a polynomial equation, the roots which define possible reflection point locations. The true point is singled out using the laws of reflection.

FLOW DIAGRAM



SYMBOL DICTIONARY

CA	COMPLEX COEFFICIENTS OF SIXTH ORDER POLYNOMIAL EQUATION
RI	ROOTS OF POLYNOMIAL EQUATION
S	SMALLEST DISTANCE FROM SOURCE TO REFLECTION POINT TO OBSERVATION POINT
SM	DISTANCE FROM SOURCE TO REFLECTION POINT PLUS THE DISTANCE FROM THE REFLECTION POINT TO THE OBSERVATION POINT
VM	ELL ANGLE DEFINING POSSIBLE REFLECTION POINT ON CYL
VIA	NORMALIZATION CONSTANT FOR VI
XR	X,Y,Z COMPONENTS OF REFLECTION POINT LOCATION ON CYLINDER

CODE LISTING

```

1 C-----
2 SUBROUTINE RFDIN(VR,UR,VI,XC)
3 C!!!
4 C!!! DETERMINES THE NEAR FIELD REFLECTION POINT FROM AN
5 C!!! ELLIPTIC CYLINDER
6 C!!!
7 COMPLEX CA(7),RT(6)
8 DIMENSION XR(3),VI(3),XC(3)
9 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
10 COMMON/SORINF/XS(3),VXS(3,3)
11 C!!! DETERMINE COEFFICIENTS OF POLYNOMIAL EQUATION
12 CA(7)=(A*A-B*B)*CMPLX(A*(XC(2)+XS(2)),B*(XC(1)+XS(1)))
13 CA(6)=-2.*CMPLX((A*A+B*B)*(XS(1)*XC(2)+XS(2)*XC(1))
14 2,2.*A*B*(A*A-B*B+XS(1)*XC(1)-XS(2)*XC(2)))
15 CA(5)=CMPLX(A*(5.*B*B-A*A)*(XS(2)+XC(2))
16 2,B*(5.*A*A-B*B)*(XS(1)+XC(1)))
17 CA(4)=CMPLX(4.*(A*A-B*B)*(XS(1)*XC(2)+XS(2)*XC(1)),0.)
18 CA(3)=CONJG(CA(5))
19 CA(2)=CONJG(CA(6))
20 CA(1)=CONJG(CA(7))
21 C!!! SOLVE POLYNOMIAL EQUATION TO OBTAIN POSSIBLE
22 C!!! REFLECTION POINTS
23 CALL POLYRT(6,CA,RT)
24 VR=BTAN2(AIMAG(RT(1)),REAL(RT(1)))
25 S=SQRT((A*COS(VR)-XS(1))**2+(B*SIN(VR)-XS(2))**2)
26 S=S+SQRT((XC(1)-A*COS(VR))**2+(XC(2)-B*SIN(VR))**2)
27 C!!! DETERMINE PHYSICALLY CORRECT REFLECTION POINTS
28 C!!! (SATISFY LAW OF REFLECTION)
29 DO 10 I=2,6
30 VM=BTAN2(AIMAG(RT(I)),REAL(RT(I)))
31 XR(1)=A*COS(VM)
32 XR(2)=B*SIN(VM)
33 SMA=(XR(1)-XS(1))*(XR(1)-XS(1))+(XR(2)-XS(2))*(XR(2)-XS(2))
34 SMB=(XC(1)-XR(1))*(XC(1)-XR(1))+(XC(2)-XR(2))*(XC(2)-XR(2))
35 SM=SQRT(SMA)+SQRT(SMB)
36 IF(S.LE.SM) GO TO 10
37 S=SM
38 VR=VM
39 10 CONTINUE
40 SNV=SIN(VR)
41 CSV=COS(VR)
42 XR(1)=A*CSV
43 XR(2)=B*SNV
44 SNX=B*CSV
45 SNY=A*SNV
46 SIX=XR(1)-XS(1)
47 SIY=XR(2)-XS(2)
48 VI(1)=XC(1)-XR(1)
49 VI(2)=XC(2)-XR(2)
50 SND=SNX*VI(1)+SNY*VI(2)
51 SNI=SNX*SIX+SNY*SIY
52 XR(3)=(SND*XS(3)-SNI*XC(3))/(SND-SNI)
53 UR=XR(3)
54 VI(3)=XC(3)-XR(3)
55 VIM=SQRT(VI(1)*VI(1)+VI(2)*VI(2)+VI(3)*VI(3))
56 DO 20 N=1,3
57 20 VI(N)=VI(N)/VIM
58 RETURN
59 END

```

RFDFPT

PURPOSE

To compute the ray path for a source ray which is reflected by the cylinder and then diffracted by a given edge on a given plate.

PERTINENT GEOMETRY

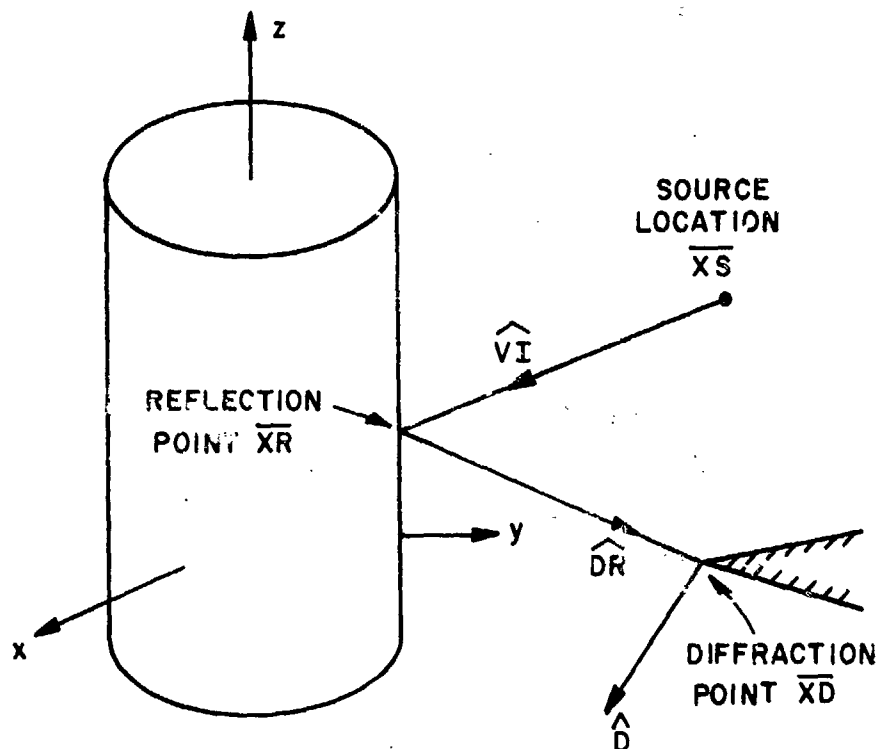


Figure 93--Illustration of ray reflected from cylinder and then diffracted by a plate edge.

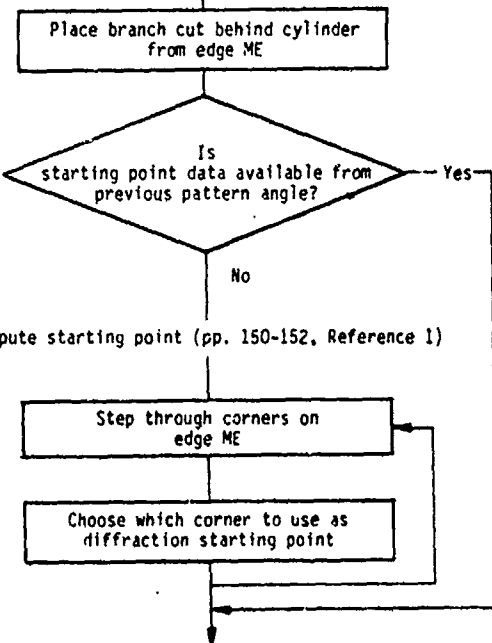
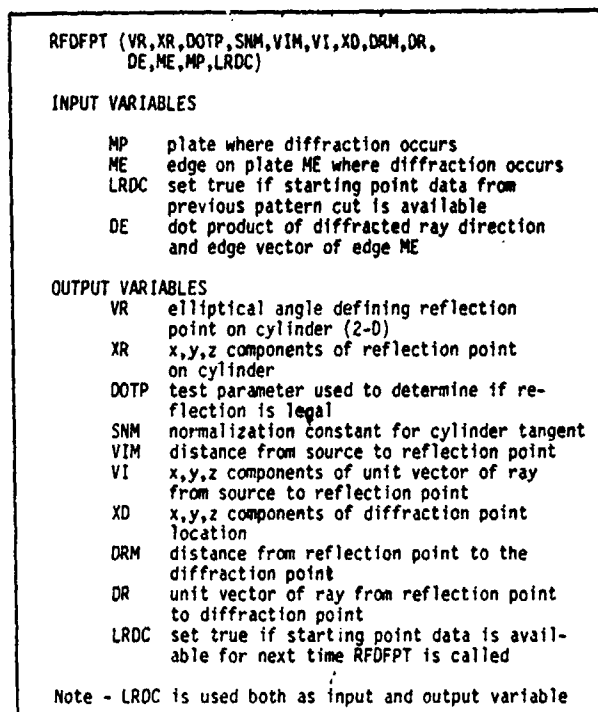
$$\overline{XR} = \hat{x} \text{ XR}(1) + \hat{y} \text{ XR}(2) + \hat{z} \text{ XR}(3)$$

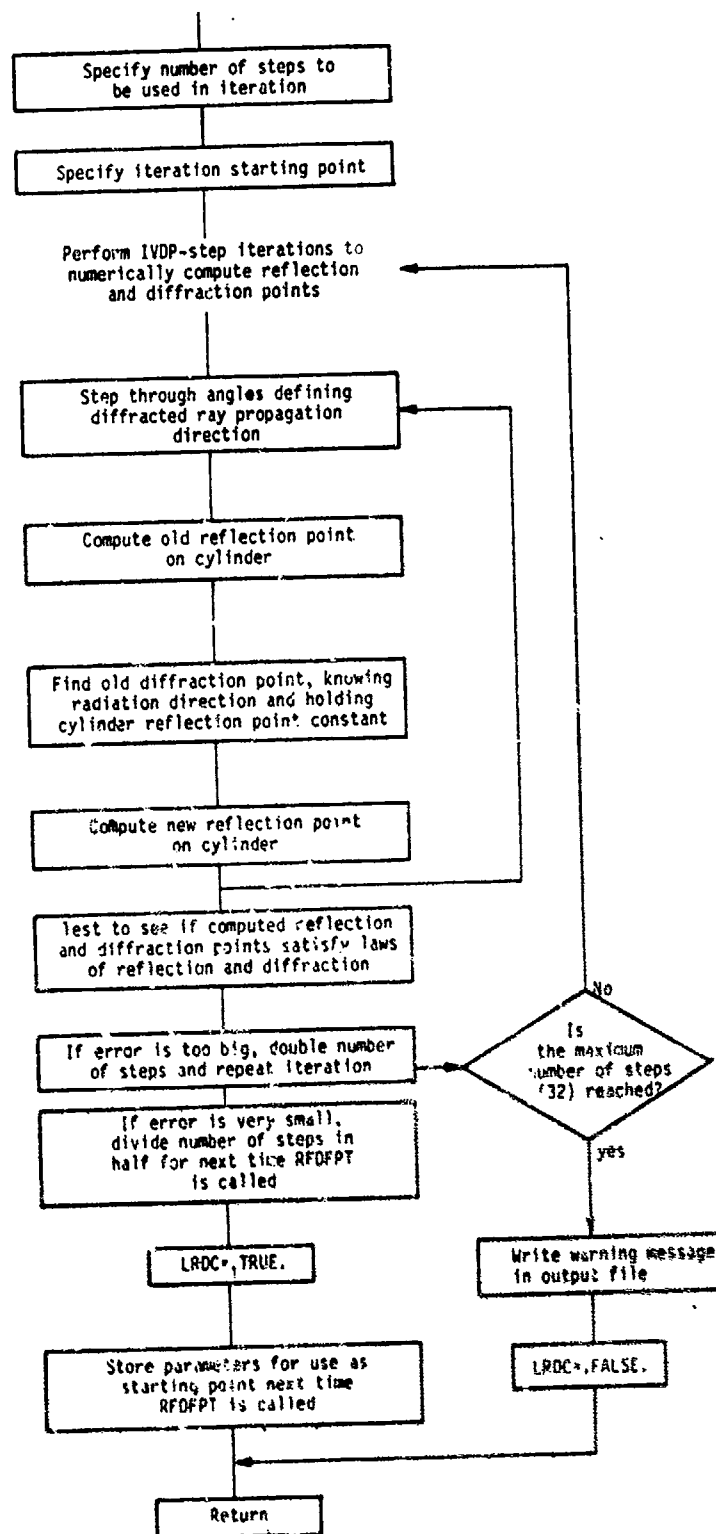
$$\overline{XD} = \hat{x} \text{ XD}(1) + \hat{y} \text{ XD}(2) + \hat{z} \text{ XD}(3)$$

METHOD

The reflection point on an elliptic cylinder and the diffraction point on a plate edge for the reflected-diffracted ray in a given observation direction is calculated via an iterative process. The equations are based on a first order Taylor series approximation to the equations governing the laws of reflection and diffraction. The details of the analysis are given on pages 141-148 of Reference 1. The iteration process follows the same basic scheme outlined in the write up for subroutine RFPTCL. The initial start up procedure for this subroutine is composed of locating the reflection point on the cylinder for a known diffraction point which is taken to be on the corners of the plate edge under consideration. The details of this procedure are discussed on pages 149-154 of Reference 1.

FLOW DIAGRAM





SYMBOL DICTIONARY

DC	X,Y,Z COMPONENTS OF DIFFRACTED RAY PROPAGATION DIRECTION USED IN ITERATION
DCP	X,Y COMPONENTS OF PHI POLARIZATION UNIT VECTOR FOR DIFFRACTED RAY USED IN ITERATION
DCT	X,Y,Z COMPONENTS OF THETA POLARIZATION UNIT VECTOR FOR DIFFRACTED RAY USED IN ITERATION
DPSR	PHI ANGLE INCREMENT SIZE
DR	X,Y,Z COMPONENTS OF RAY DIRECTION BETWEEN REFLECTION AND DIFFRACTION
DRP	PARTIAL DERIVATIVE OF DR WITH RESPECT TO PHI
DRT	PARTIAL DERIVATIVE OF DR WITH RESPECT TO THETA
DRU	PARTIAL DERIVATIVE OF DR WITH RESPECT TO UR
DRV	PARTIAL DERIVATIVE OF DR WITH RESPECT TO VR
DTSR	THETA ANGLE INCREMENT SIZE
DU	CHANGE IN UR FOR ONE ITERATION USING TAYLOR SERIES EXPANSION
DV	CHANGE IN VR FOR ONE ITERATION USING TAYLOR SERIES EXPANSION
ERC	ERROR DETECTION VARIABLE
FI	EQUATION GOVERNING THE LAW OF REFLECTION
FP	PARTIAL DERIVATIVE OF FI WITH RESPECT TO PHI
FT	PARTIAL DERIVATIVE OF FI WITH RESPECT TO THETA
FU	PARTIAL DERIVATIVE OF FI WITH RESPECT TO UR
FV	PARTIAL DERIVATIVE OF FI WITH RESPECT TO VR
GI	EQUATION GOVERNING THE LAW OF REFLECTION
GP	PARTIAL DERIVATIVE OF GI WITH RESPECT TO PHI
GT	PARTIAL DERIVATIVE OF GI WITH RESPECT TO THETA
GU	PARTIAL DERIVATIVE OF GI WITH RESPECT TO UR
GV	PARTIAL DERIVATIVE OF GI WITH RESPECT TO VR
IVD	STORED NUMBER OF STEPS USED IN ITERATION
LKDC	SET TRUE IF STARTING POINT DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE
PHCR	PHI COMPONENT OF DIFFRACTED RAY DIRECTION USED IN ITERATION
PHOR	PHI COMPONENT OF DIFFRACTED RAY DIRECTION FROM PREVIOUS TIME REFDEPT WAS CALLED (OR PRESENT VALUE FOR NEXT TIME ROUTINE IS CALLED)
PHCRP	PHI ANGLE OF DIFFRACTED RAY DIRECTION IN ROTATED RCS SYSTEM (BRANCH CUT PLACED BEHIND CYL)
PHSPR	PHI ANGLE OF DIFFRACTED RAY DIRECTION IN ROTATED RCS SYSTEM (BRANCH CUT PLACED BEHIND CYLINDER)
SNPX	PARTIAL DERIVATIVE OF SNX WITH RESPECT TO ANGLE VR
SNPY	PARTIAL DERIVATIVE OF SNY WITH RESPECT TO ANGLE VR
SNX	X COMPONENT OF NORMAL TO CYLINDER
SNY	Y COMPONENT OF NORMAL TO CYLINDER
STP	NUMBER OF STEPS USED IN ITERATION
THCR	THETA COMPONENT OF DIFFRACTED RAY DIRECTION USED IN ITERATION
THOR	THETA COMPONENT OF DIFFRACTED RAY DIRECTION FROM PREVIOUS TIME REFDEPT WAS CALLED (OR FOR NEXT TIME ROUTINE IS CALLED)
UR	Z COMPONENT OF REFLECTION POINT LOCATION ON CYLINDER
URC	STORED COMPONENTS DEFINING Z COMPONENT OF STARTING REFLECTION POINT LOCATIONS ON CYLINDER
VI	X,Y,Z COMPONENTS OF DIRECTION OF RAY INCIDENT ON CYLINDER
VIC	PARTIAL DERIVATIVE OF VI WITH RESPECT TO UR
VIV	PARTIAL DERIVATIVE OF VI WITH RESPECT TO ANGLE VR
VN	ELL ANGLE DEFINING REFLECTION POINT ON CYLINDER
VNC	STORED ELL ANGLES DEFINING STARTING REFLECTION POINT LOCATIONS ON CYLINDER
XD	X,Y,Z COMPONENTS OF DIFFRACTION POINT LOCATION
XR	X,Y,Z COMPONENTS OF REFLECTION POINT LOCATION ON CYLINDER

CODE LISTING

```

1 C-----
2      SUBROUTINE NFDPT(VR,XR,DOTP,SHM,VIA,VI,XD,DRM,DR,DE,ME,MP
3      2,LRDC)
4 C!!!
5 C!!! DETERMINES THE RAY PATH FOR A REFLECTION FROM THE ELLIPTIC
6 C!!! CYLINDER THEN DIFFRACTION FROM A PLATE EDGE
7 C!!!
8      DIMENSION DC(3),DCP(2),DCT(3),VI(3),VIV(3),VIU(3),VSD(3)
9      DIMENSION XP(3),XR(3),XRP(3),XRV(3),XRU(3),XD(3)
10     DIMENSION DR(3),DRU(3),DRV(3),DRT(3),DRP(3)
11     DIMENSION IVD(14,6),PHOR(14,6),THOR(14,6),VRO(14,6),URO(14,6)
12     DIMENSION PHCRP(14,6)
13     LOGICAL LRFC
14     COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
15     2,MEP(14),MPX
16     COMMON/SORIP/XS(3),VXS(3,3)
17     COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
18     COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
19     COMMON/LNDRCL/VCD(14,6),UCD(14,6),BCD(14,6,2)
20     COMMON/RNPPH/PHWR(14,6)
21     COMMON/PIS/PI,TPI,DPR,RPD
22 C!!! PLACE BRANCH CUT BEHIND CYLINDER FROM EDGE
23     PHSPR=PHSR-PHWR(MP,ME)
24     IF(PHSPR.GT.PI) PHSPR=PHSPR-TPI
25     IF(PHSPR.LT.-PI) PHSPR=PHSPR+TPI
26 C!!! IS STARTING POINT DATA AVAILABLE FROM
27 C!!! PREVIOUS PATTERN ANGLE?
28     IF(LRDC) GO TO 40
29 C!!! COMPUTE STARTING POINT
30     DOX=-2.
31 C!!! STEP THRU CORNERS ON EDGE ME
32 C!!! CHOOSE WHICH CORNER TO USE AS STARTING
33 C!!! DIFFRACTION POINT
34     DO 5 J=1,2
35     MC=ME-1+J
36     IF(LC.GT.MEP(MP)) MC=1
37     ISGN=1
38     SA=SQRT((1.-DE*DE)/(1.-BCD(MP,ME,J)+BCD(MP,ME,J)))
39     AP=-DE+ISGN*SA*BCD(MP,ME,J)
40     DAX=D(1)+AP*V(MP,ME,1)
41     DAY=D(2)+AP*V(MP,ME,2)
42     DAZ=D(3)+AP*V(MP,ME,3)
43     SA=DAX*DAX+DAY*DAY
44     SB=SA+DAZ*DAZ
45     SA=SQRT(SA)
46     SB=SQRT(SB)
47     CPOP=DAX/SA
48     SPOP=DAY/SA
49     CTCR=DAZ/SB
50     STCR=SA/SB
51     DOX=CPOP*STOP
52     DOY=SPOP*STOP
53     DOZ=CTCR
54     DOO=D(1)+DOX+D(2)+DOY+D(3)+DOZ
55     DOV=DOX*V(MP,ME,1)+DOY*V(MP,ME,2)+DOZ*V(MP,ME,3)
56     IF(DOV+ICD(MP,ME,J).LT.0.) GO TO 4
57     IF(ABS(ICD).GT.1.) GO TO 4
58     IF(DOZ.LE.DOX) GO TO 4
59     ICR=DOO
60     JE=J
61     CPO=CPOP
62     SPO=SPOP
63     CTO=CTCR
64     STO=STCR
65     ISGN=-ISGN
66     IF(1872.LT.0) GO TO 3

```

```

67 5      CONTINUE
68      PHOR(MP,ME)=ETAN2(SPO,CPO)
69      PHORP(MP,ME)=PHOR(MP,ME)-PHWR(MP,ME)
70      IF (PHORP(MP,ME).GT.PI) PHORP(MP,ME)=PHORP(MP,ME)-TPI
71      IF (PHORP(MP,ME).LT.-PI) PHORP(MP,ME)=PHORP(MP,ME)+TPI
72      THOR(MP,ME)=ETAN2(STO,CTO)
73      MC=ME-1+JE
74      IF (MC.GT.MEP(MP)) MC=1
75      VRO(MP,ME)=VCD(MP,MC)
76      URO(MP,ME)=UCD(MP,MC)
77      IVD(MP,ME)=1
78 C!!!   SPECIFY NUMBER OF STEPS IN ITERATION
79      STP=IVD(MP,ME)
80      IVDP=IVD(MP,ME)+1
81      DPSR=(PFSR-PHORP(MP,ME))/STP
82      DTSR=(TFSR-THOR(MP,ME))/STP
83 C!!!   SPECIFY STARTING POINT
84      VR=VRO(MP,ME)
85      UR=URO(MP,ME)
86 C!!!   PERFORM IVDP-STEP ITERATIONS TO NUMERICALLY
87 C!!!   COMPUTE REFLECTION AND DIFFRACTION POINTS.
88 C!!!   STEP THROUGH ANGLES (DEFINING DIF. RAY PROP. DIR.)
89      DO 52 IV=1,IVDP
90      PHCR=PHOR(MP,ME)+(IV-1)*DPSR
91      THCR=THOR(MP,ME)+(IV-1)*DTSR
92      CPCS=COS(PHCR)
93      SPCS=SIN(PHCR)
94      CTCR=COS(THCR)
95      STCR=SIN(THCR)
96      DC(1)=CPCS*STCR
97      LC(2)=SPCS*STCR
98      DC(3)=CTCR
99      PCP(1)=-SPCS*STCR
100     PCP(2)=CPCS*STCR
101     DCT(1)=CPCS*CTCR
102     DCT(2)=SPCS*CTCR
103     DCT(3)=-STCR
104     CSV=COS(VR)
105     SNV=SIN(VR)
106     SNX=B*CSV
107     SNY=A*SNV
108     SNPX=-B*SNV
109     SNPY=A*CSV
110 C!!!   COMPUTE OLD REFLECTION POINT ON CYLINDER
111     XR(1)=A*CSV
112     XR(2)=B*SNV
113     XR(3)=UR
114     XRV(1)=-A*SNV
115     XRV(2)=B*CSV
116     XRV(3)=C.
117     XRU(1)=C.
118     XRU(2)=0.
119     XRU(3)=1.
120     PV=0.
121     DDV=0.
122     DO 10 N=1,3
123     VI(N)=XR(N)-XS(N)
124     DDV=DDV+DC(N)*V(MP,ME,N)
125 10     PV=PV+(XR(N)-X(MP,ME,N))*V(MP,ME,N)
126     DO 11 N=1,3
127 11     XP(N)=X(MP,ME,N)+PV*V(MP,ME,N)
128     SM=0.
129     DO 12 N=1,3
130 12     SM=SM+(XR(N)-XP(N))*(XR(N)-XP(N))
131     SR=SQRT(SM)
132     COLE=DDV/SQRT(1.-DDV*DDV)

```

```

133 C!!! FIND OLD DIFFRACTION POINT, KNOWING RADIATION
134 C!!! DIRECTION AND HOLDING CYLINDER REFLECTION
135 C!!! POINT CONSTANT
136 DO 13 N=1,3
137 XD(N)=XP(N)+SM*COTB*V(MP,ME,N)
138 DR(N)=XD(N)-XR(N)
139 VIV(N)=XRV(N)
140 13 VIU(N)=XRU(N)
141 IF(IV.EQ.IVDP) GO TO 60
142 DDPV=DCF(1)*V(MP,ME,1)+DCP(2)*V(MP,ME,2)
143 DDIV=DCT(1)*V(MP,ME,1)+DCT(2)*V(MP,ME,2)+DCT(3)*V(MP,ME,3)
144 DDDV=(1.-DDV*DDV)**1.5
145 CTBP=DDFV/DDIV
146 CTBT=DDIV/DDDV
147 DO 14 N=1,3
148 DRP(N)=SM*CTBP*V(MP,ME,N)
149 14 DRT(N)=SM*CTBT*V(MP,ME,N)
150 CRUV=0.
151 CRVV=0.
152 CRUR=0.
153 CRVR=0.
154 CRV=0.
155 DO 15 N=1,3
156 CRUV=CRUV+XRU(N)*V(MP,ME,N)
157 CRVV=CRVV+XRV(N)*V(MP,ME,N)
158 CRUR=CRUR+XRU(N)*(XR(N)-X(MP,ME,N))
159 15 CRVR=CRVR+XRV(N)*(XR(N)-X(MP,ME,N))
160 CCU=CRUV+COTB*(CRUR-CRUV*PV)/SM
161 CCV=CRVV+COTB*(CRVR-CRVV*PV)/SM
162 DO 16 N=1,3
163 DRU(N)=CCU*V(MP,ME,N)-XRU(N)
164 16 DRV(N)=CCV*V(MP,ME,N)-XRV(N)
165 C!!! PERFORM TAYLOR SERIES EXPANSION TO DEFINE DV AND DU
166 FV=(SNPX*VI(1)+SNX*VIV(1)+SNPY*VI(2)+SNY*VIV(2))*
167 2(SNX*DR(2)-SNY*DR(1))
168 FV=FV+(SNPX*DR(2)+SNX*DRV(2)-SNPY*DR(1)-SNY*DRV(1))*
169 2(SNX*VI(1)+SNY*VI(2))
170 FV=FV+(SNPX*VI(2)+SNX*VIV(2)-SNPY*VI(1)-SNY*VIV(1))*
171 2(SNX*DR(1)+SNY*DR(2))
172 FV=FV+(SNPX*DR(1)+SNX*DRV(1)+SNPY*DR(2)+SNY*DRV(2))*
173 2(SNX*VI(2)-SNY*VI(1))
174 FU=(SNX*DR(2)-SNY*DR(1))*(SNX*VIU(1)+SNY*VIU(2))+
175 2(SNX*DR(1)+SNY*DR(2))*(SNX*VIU(2)-SNY*VIU(1))
176 FU=FU+(SNX*VI(1)+SNY*VI(2))*(SNX*DRU(2)-SNY*DRU(1))+
177 2(SNX*DRU(1)+SNY*DRU(2))*(SNX*VI(2)-SNY*VI(1))
178 GV=DR(3)*(SNPX*VI(1)+SNX*VIV(1)+SNPY*VI(2)+SNY*VIV(2))
179 GV=GV+VI(3)*(SNPX*DR(1)+SNX*DRV(1)+SNPY*DR(2)+SNY*DRV(2))
180 GV=GV+DRV(3)*(SNX*VI(1)+SNY*VI(2))+VIV(3)*(SNX*DR(1)+SNY*DR(2))
181 GU=DR(3)*(SNX*VIU(1)+SNY*VIU(2))+VIU(3)*(SNX*DR(1)+SNY*DR(2))
182 GU=GU+DRU(3)*(SNX*VI(1)+SNY*VI(2))+VI(3)*(SNX*DRU(1)+SNY*DRU(2))
183 FP=(SNX*VI(1)+SNY*VI(2))*(SNX*DRP(2)-SNY*DRP(1))+
184 2(SNX*VI(2)-SNY*VI(1))*(SNX*DRP(1)+SNY*DRP(2))
185 FT=(SNX*VI(1)+SNY*VI(2))*(SNX*DRT(2)-SNY*DRT(1))+
186 2(SNX*VI(2)-SNY*VI(1))*(SNX*DRT(1)+SNY*DRT(2))
187 GP=VI(3)*(SNX*DRP(1)+SNY*DRP(2))+DRP(3)*(SNX*VI(1)+SNY*VI(2))
188 GT=DRT(3)*(SNX*VI(1)+SNY*VI(2))+VI(3)*(SNX*DRT(1)+SNY*DRT(2))
189 FI=(SNX*VI(1)+SNY*VI(2))*(SNX*DR(2)-SNY*DR(1))+
190 2(SNX*DR(1)+SNY*DR(2))*(SNX*VI(2)-SNY*VI(1))
191 GI=DR(3)*(SNX*VI(1)+SNY*VI(2))+VI(3)*(SNX*DR(1)+SNY*DR(2))
192 DE F=FU*GV-FV*GU
193 DV=((FI*GU-GI*FU)+(GU*FP-FU*GP)*DPSR+(GU*FT-FU*GT)*DTSR)/DET
194 DU=((GI*FV-FI*GV)+(FV*GP-GV*FP)*DPSR+(FV*GT-GV*FT)*DTSR)/DET
195 C!!! COMPUTE NEW REFLECTION POINT ON CYLINDER
196 UR=UR+DU
197 40 VR=VR+DV
198 50 CONTINUE

```

```

199 GO      CONTINUE
200 C!!!    TEST TO SEE IF COMPUTED SCATTER POINTS SATISFY
201 C!!!    LAWS OF REFLECTION AND DIFFRACTION
202          SNM=SQRT(SNX*SNX+SNY*SNY)
203          SNX=SNX/SNM
204          SNY=SNY/SNM
205          DDV=0.
206          DRM=0.
207          VIM=0.
208          DO 20 N=1,3
209             VIM=VIM+VI(N)*VI(N)
210             DDRV=DDRV+DR(N)*V(MP,ME,N)
211 20       DRM=DRM+DR(N)*DR(N)
212          VIM=SQRT(VIM)
213          DRM=SQRT(DRM)
214          DO 30 N=1,3
215             VI(N)=VI(N)/VIM
216 30       DR(N)=DR(N)/DRM
217          DDRV=DDRV/DRM
218          ERCB=ABS(DDV-DDRV)
219          SHAD=SNX*DR(1)+SNY*DR(2)
220          SHADC=SNX*VI(1)+SNY*VI(2)
221          ERC=SHAD+SHADC
222          DUTP=.5*(SHAD-SHADC)
223          ERCA=ABS(ERC)
224          ERC=ERCA
225          IF(ERCB.GT.ERC)ERC=ERCB
226 C!!!    IF ERROR IS VERY SMALL, DIVIDE NUMBER OF STEPS
227 C!!!    IN HALF FOR NEXT TIME ROUTINE IS CALLED
228          IF(ERC.LT.0.01) GO TO 80
229 C!!!    IF ERROR IS TOO BIG, DOUBLE NUMBER OF STEPS
230 C!!!    (UP TO 32) AND REPEAT ITERATION
231          IF(IVD(MP,ME).GE.32) GO TO 70
232          IVD(MP,ME)=2*IVD(MP,ME)
233          GO TO 40
234 70       CONTINUE
235          WRITE(6,1) PHSR,THSR,MP,ME,VR,UR,ERCA,ERCB
236 1         FORMAT(' ERROR IN RFDFT= ',2F12.6,2I5,4F12.6)
237          LRDC=.FALSE.
238          RETURN
239 80       CONTINUE
240          IF(ERC.GE.0.01) GO TO 90
241          IF(IVD(MP,ME).EQ.1) GO TO 90
242          IVD(MP,ME)=IVD(MP,ME)/2
243 90       CONTINUE
244 C!!!    STORE PARAMETERS FOR NEXT TIME RFDFT IS CALLED
245          VR(MP,ME)=VR
246          UR(MP,ME)=UR
247          PHOR(MP,ME)=PHSR
248          PHORP(MP,ME)=PHSPH
249          THOR(MP,ME)=THSR
250          IF(.NOT.LRDC) IVD(MP,ME)=1
251          LRDC=.TRUE.
252          RETURN
253          END

```

REFPTCL

PURPOSE

To calculate the reflection point on the elliptic cylinder for a source ray reflected in a given direction. The routine also computes cylinder reflection points for source rays that are reflected by a given plate and then reflected by the cylinder.

PERTINENT GEOMETRY

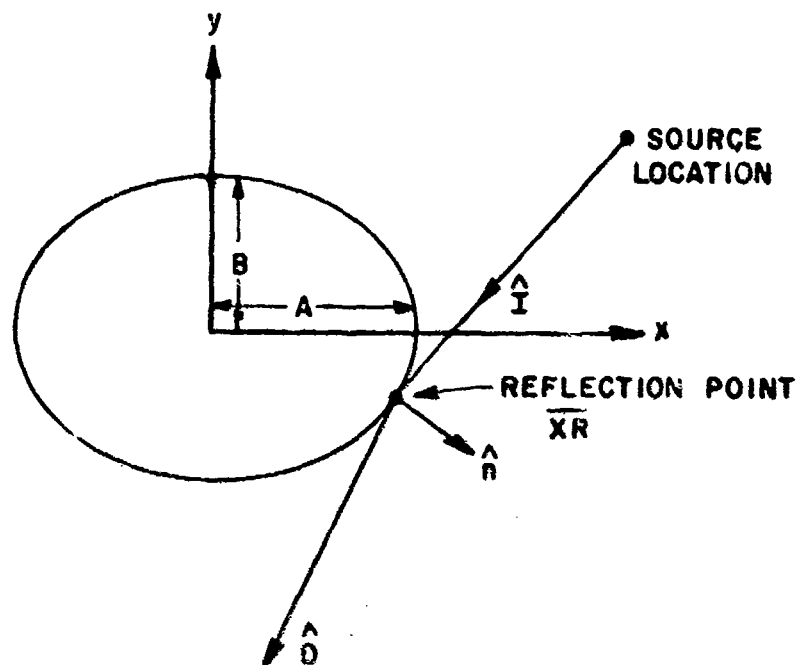


Figure 94-- Illustration of cylinder reflection point.

$$\hat{I} = \hat{x} \sin X + \hat{y} \sin Y$$

$$\hat{n} = \hat{x} \sin X + \hat{y} \sin Y$$

$$\overline{XR} = \hat{x} A \cos VR + \hat{y} B \sin VR$$

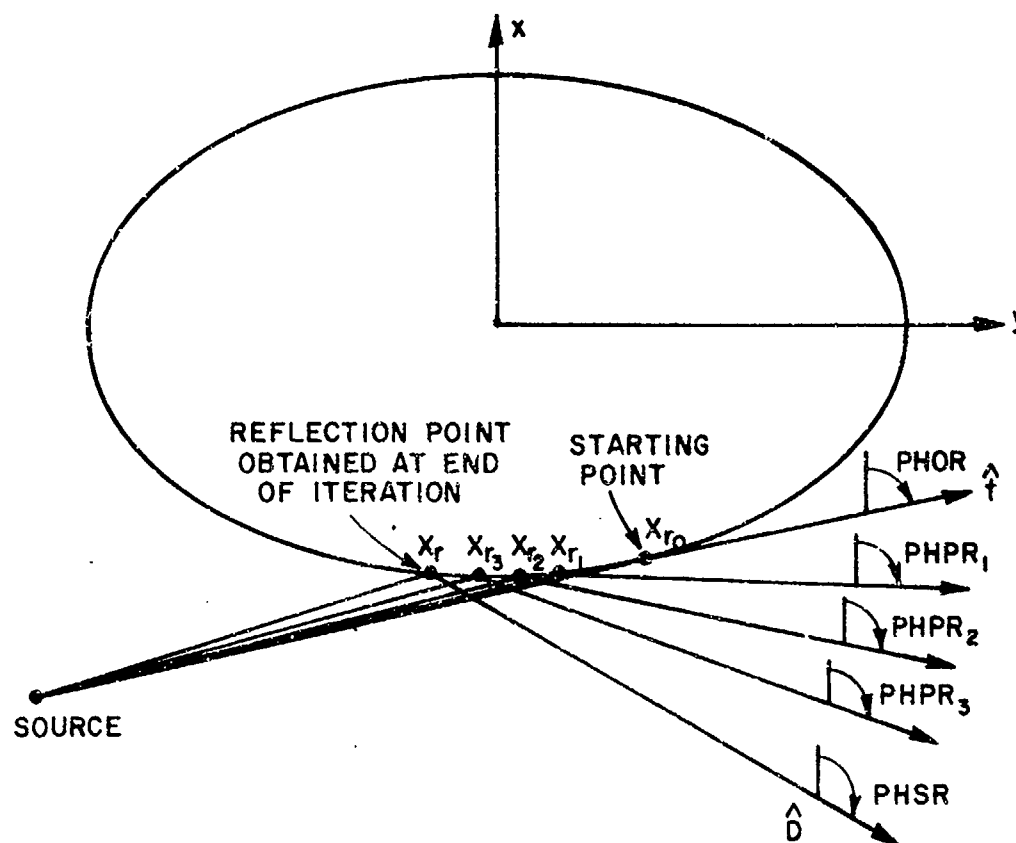


Figure 96--Illustration of iterative method used in computing the cylinder reflection point.

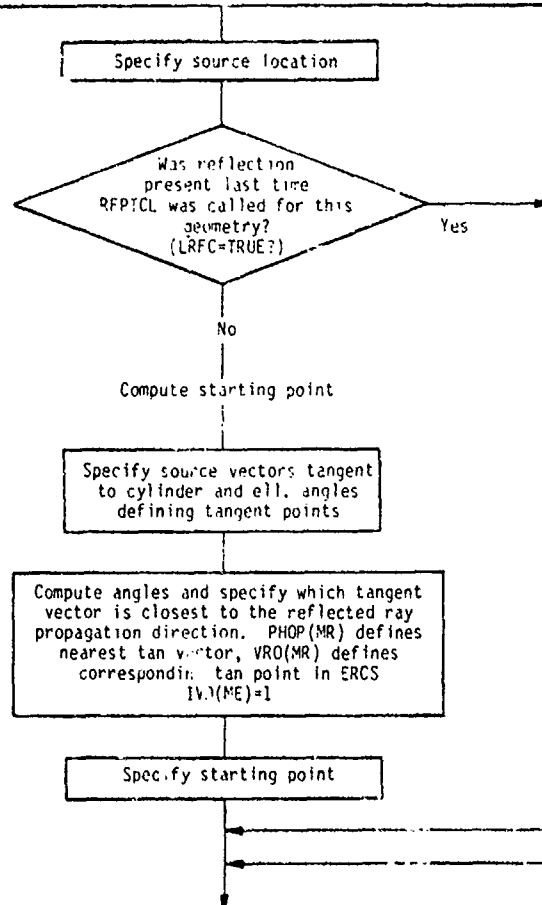
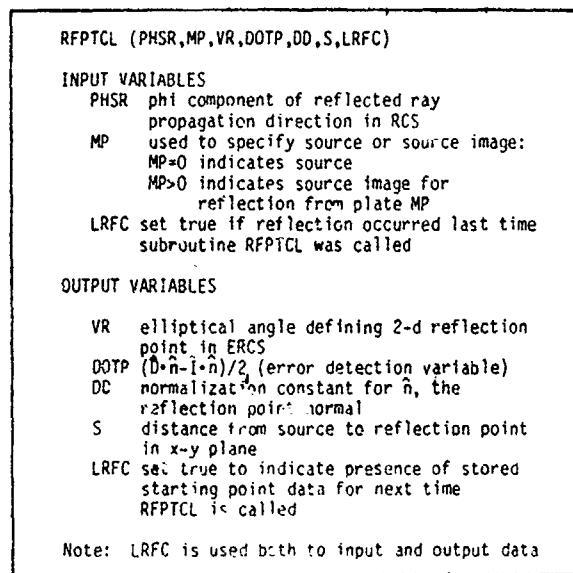
METHOD

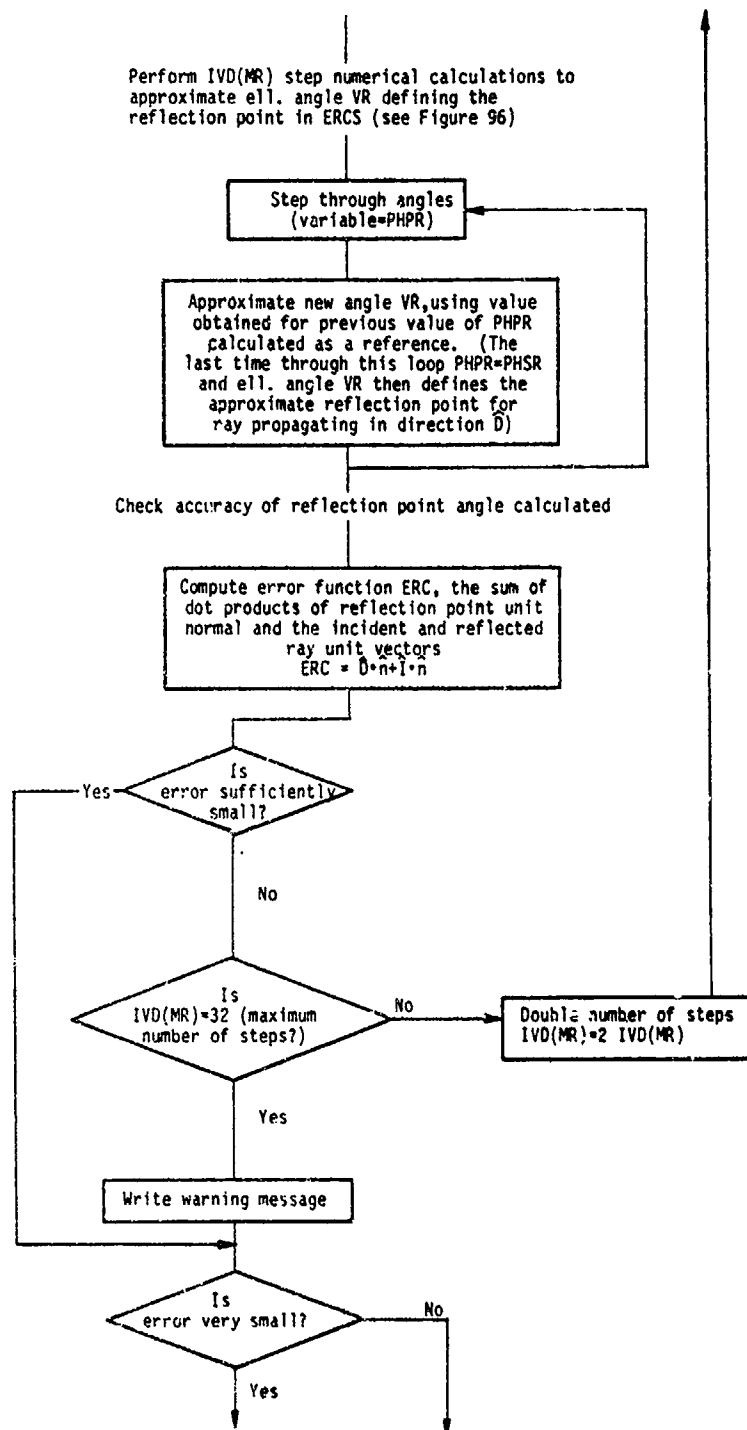
The reflection point for a ray reflected in a direction defined by the phi angle PHSR is calculated via an iterative process. The routine starts with the tangent ray nearest to the reflected ray direction (or other nearby reflected ray whose reflection point is known) and steps along the cylinder surface, calculating the approximate reflection point for each reflected ray phi angle PHPR (which is stepped from PHOR to PHSR in evenly spaced steps). Each reflection point calculation uses the previous reflection point as a reference. As long as the steps are sufficiently small, the approximation is accurate. The equations are based on a first order Taylor series approximation of the equation governing the laws of reflection. Further details are given on pages 102-104 of Reference 1. The point obtained at the end of the process is the estimated reflection point. The routine then takes the sum of dot products of the cylinder normal and the incident

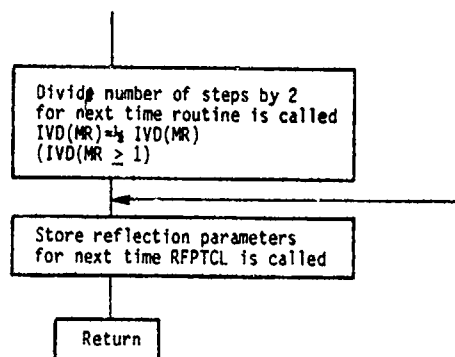
and reflected rays (which should be zero in order to satisfy the law of reflection). If it is larger than some minimal amount, the number of steps used to iterate angle PHPR is doubled and the calculation is done again. If the error is much smaller than necessary, the number of steps used in the next calculation is divided by two.

Once a reflection point is calculated for a particular geometry, the elliptical angle defining the reflection point (VRO(MR)) is saved, along with the number of steps used to calculate it (IVD(MR)) for the next time RFPTCL is called for the same geometry. Since the next pattern angle is likely to be quite close to the previous one, this gives the computer a good starting point in defining the next reflection point, hence minimizing computer time. LRFC is a logical variable which if true tells the user that there is data from the previous pattern angle available to compute the next reflection point. If a reflection does not occur, LRFC is set false, and the next time the routine is called, it will start at the nearest tangent point.

FLOW DIAGRAM







SYMBOL DICTIONARY

CPP	COSINE OF PHPR
CPS	COSINE OF PHSR
CSV	COSINE OF VR
DD	NORMALIZATION CONSTANT FOR REFL POINT NORMAL VECTOR
DOTP	ONE HALF THE DIFFERENCE BETWEEN THE DOT PRODUCTS OF THE REFLECTED RAY DIRECTION AND CYLINDER UNIT NORMAL AND THE INCIDENT RAY DIRECTION AND CYLINDER UNIT NORMAL
DPX }	X AND Y COMPONENTS OF PARTIAL DERIVATIVE OF REFLECTED RAY
DPY }	DIRECTION WITH RESPECT TO PHI OBSERVATION ANGLE
DR	DOT PRODUCT OF INC RAY UNIT VECTOR AND CYL UNIT NORMAL
DS	DOT PRODUCT OF REFLECTED RAY PROPAGATION DIRECTION UNIT VECTOR AND CYLINDER UNIT NORMAL
DSPH	SIZE OF ANGLE STEP USED IN ITERATION
DV	CHANGE IN ANGLE VR
DX }	X AND Y COMPONENTS OF UNIT VECTOR OF REFLECTED RAY
DY }	(DIRECTION DEFINED BY ANGLE PHPR) IN RCS
DVB	PARTIAL DERIVATIVE OF THE REFLECTION LAW EQUATION (FI) WITH RESPECT TO ELL ANGLE V
DVT	PARTIAL DERIVATIVE OF THE REFLECTION LAW EQUATION (FI) WITH RESPECT TO THE PHI ANGLE OF THE OBSERVATION DIRECTION
ERC	ERROR PARAMETER (SUM OF DS AND DR)
ERCA	ABSOLUTE VALUE OF ERC
ERCS	(NOT A VARIABLE) ABBREVIATION FOR ELLIPTICAL REFERENCE COORDINATE SYSTEM
FI	EQUATION SATISFYING THE LAW OF REFLECTION
IVD	NUMBER OF ITERATIONS USED TO FIND REFL POINT THE LAST TIME REPTCL WAS CALLED FOR PLATE MP
IVDM	NUMBER OF STEPS USED IN ITERATION
LRFC	(ENTERING ROUTINE) SET TRUE IF REFL OCCURED LAST TIME REFCYL WAS CALLED. (LRFC ALWAYS SET TRUE LEAVING ROUTINE)
MP	USED TO SPECIFY WHETHER SOURCE OR SOURCE IMAGE IS USED MP=0 DESIGNATES SOURCE MP>0 DESIGNATES SOURCE IMAGE FOR REFLECTION FROM PLATE MP
MR	INDEX VARIABLE (MP+MPRX+1) FOR STORING DATA FOR NEXT CALL TO REPTCL
PHE	PHI ANGLE BETWEEN REFLECTED RAY DIRECTION AND TANGENT POINT #2
PHEP	PHI ANGLE BETWEEN REFLECTED RAY DIRECTION AND TANGENT POINT #1
PHIR	PHI COMPONENT OF SOURCE LOCATION IN RCS
PHOR	REFLECTED RAY PHI ANGLE (STORED AS STARTING POINT PARAMETER FOR NEXT TIME ROUTINE IS CALLED)
PHORB	PHI ANGLE DEFINING RAY TANGENT TO TAN POINT 1
PHORP	PHI ANGLE OF CYLINDER REFLECTED RAY DIRECTION IN ROTATED RCS SYSTEM
PHPR	REFLECTED RAY PHI ANGLE (ITERATED FROM PHOR TO PHSR)
PHSPR	PHI ANGLE DEFINING REFLECTED RAY DIRECTION IN ROTATED RCS
PHSR	PHI COMPONENT OF REFLECTED RAY PROPAGATION DIRECTION IN RCS
S	DISTANCE FROM SOURCE TO REFL POINT IN X-Y PLANE
SIPX }	X AND Y COMPONENTS OF PARTIAL DERIVATIVE OF INCIDENT
SIPY }	RAY VECTOR WITH RESPECT TO ELL ANGLE V
SIX }	X AND Y COMPONENTS OF INCIDENT RAY PROP VECTOR
SIY }	IN RCS (NOT CONSISTANTLY NORMALIZED)
SNPX }	X AND Y COMPONENTS OF PARTIAL DERIVATIVE OF CYLINDER
SNPY }	NORMAL AT REFLECTION POINT WITH RESPECT TO ELL ANGLE V
SNV	SINE OF VR
SNX }	X AND Y COMPONENTS OF RAY NORMAL TO CYL REFL POINT
SNY }	IN RCS (NOT CONSISTANTLY NORMALIZED)
SPP	SINE OF PHPR
SPS	SINE OF PHSR
STP	NUMBER OF STEPS USED IN ITERATION

VR ELL. ANGLE DEFINING REFL POINT IN ERCS
VRO ELL ANGLES DEFINING TANGENT POINTS FOR SOURCE RAY (OR
XIS SOURCE RAY REFLECTED FROM PLATE) TANGENT TO CYLINDER
SOURCE LOCATION

CODE LISTING

```

1 C-----
2 SUBROUTINE RFPTCL(PHSR,MP,VR,DOTP,DD,S,LRFC)
3 C!!!
4 C!!! DETERMINES REFLECTION POINT ON AN ELLIPTIC CYLINDER
5 C!!!
6 LOGICAL LRFC,LGRND
7 DIMENSION IVD(29),PHOR(29),VRO(29),XIS(3),PHORP(29)
8 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
9 COMMON/SORINF/XS(3),VXS(3,3)
10 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
11 2,MEP(14),MPX
12 COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
13 COMMON/PIS/PI,TPI,DPR,RPD
14 COMMON/ENDSCL/DTS,VTS(2),BTS(4)
15 COMMON/ENDICL/DTI(14),VTI(14,2),BTI(14,4)
16 COMMON/GROUND/LGRND,MPXR
17 MR=MP+MPXR+1
18 SPS=SIN(PHSR)
19 CPS=COS(PHSR)
20 C!!! SPECIFY SOURCE LOCATION
21 IF(MP.GT.0) GO TO 11
22 DO 10 N=1,3
23 10 XIS(N)=XS(N)
24 PHIR=BTAN2(XS(2),XS(1))
25 GO TO 15
26 11 CONTINUE
27 DO 12 N=1,3
28 12 XIS(N)=XI(MP,MP,N)
29 PHIR=BTAN2(XI(MP,MP,2),XI(MP,MP,1))
30 15 CONTINUE
31 PHSPR=PHSR-PHIR
32 IF(PHSPR.GT.PI) PHSPR=PHSPR-TPI
33 IF(PHSPR.LT.-PI) PHSPR=PHSPR+TPI
34 C!!! WAS REFLECTION PRESENT LAST TIME REFCYL WAS CALLED?
35 IF(LRFC) GO TO 40
36 IVD(MR)=1
37 C!!! SPECIFY TANGENT VECTORS
38 IF(MP.GT.0) GO TO 20
39 PHOR(MR)=BTAN2(BTS(4),BTS(3))
40 VRO(MR)=VTS(2)
41 PHORB=BTAN2(BTS(2),BTS(1))
42 GO TO 25
43 20 CONTINUE
44 PHOR(MR)=BTAN2(BTI(MP,4),BTI(MP,3))
45 VRO(MR)=VTI(MP,2)
46 PHORB=BTAN2(BTI(MP,2),BTI(MP,1))
47 25 CONTINUE
48 C!!! COMPUTE ANGLES AND SPECIFY WHICH TAN VECTOR IS CLOSER
49 C!!! TO THE REFL PROPAGATION DIRECTION
50 PHORP(MR)=PHOR(MR)-PHIR
51 IF(PHORP(MR).GT.PI) PHORP(MR)=PHORP(MR)-TPI
52 IF(PHORP(MR).LT.-PI) PHORP(MR)=PHORP(MR)+TPI
53 PHORBP=PHORB-PHIR
54 IF(PHORBP.GT.PI) PHORBP=PHORBP-TPI
55 IF(PHORBP.LT.-PI) PHORBP=PHORBP+TPI
56 PHE=ABS(PHSPR-PHORP(MR))
57 PHEP=ABS(PHSPR-PHORBP)
58 IF(PHEP.GE.PHE) GO TO 40
59 PHOR(MR)=PHORB
60 PHORP(MR)=PHORBP
61 VRO(MR)=VTS(1)
62 IF(MP.GT.0) VRO(MR)=VTI(MP,1)
63 C!!! INCREMENT ANGLE PHPR FROM THE CYL TAN ANGLE PHOR TO
64 C!!! PROP. ANGLE PHSR IN IVD(MR) STEPS AND CALCULATE APPROX.
65 C!!! VR (THE ELL. ANGLE DEFINING THE REFL POINT) FOR EACH
66 C!!! ANGLE PHPR UNTIL PHPR=PHSR AND APPROX VR FOR REFL POINT

```

```

67 C!!! IN DESIRED PROP. DIRECTION IS OBTAINED.
68 40 STP=IVD(MR)
69 DPSR=(PHSPR-PHORP(MR))/STP
70 VR=VRO(MR)
71 IVDM=IVD(MR)
72 C!!! STEP THRU ANGLES
73 DO 50 IV=1,IVDM
74 PHPR=PHCR(MR)+(IV-1)*DPSR
75 CPP=COS(PHPR)
76 SPP=SIN(PHPR)
77 DX=CPP
78 DY=SPP
79 DPX=-SPP
80 DPY=CPP
81 CSV=COS(VR)
82 SNV=SIN(VR)
83 SNX=B*CSV
84 SNY=A*SNV
85 SIX=A*CSV-XIS(1)
86 SIY=B*SNV-XIS(2)
87 SNPX=-B*SNV
88 SNPY=A*CSV
89 SIPX=-A*SNV
90 SIY=B*CSV
91 FI=(SNX*SIX+SNY*SIY)*(SNX*DY-SNY*DX)+
92 2(SNX*DX+SNY*DY)*(SNX*SIY-SNY*SIX)
93 DVT=(SNX*SIX+SNY*SIY)*(SNX*DPY-SNY*DPX)
94 DVT=DVT+(SNX*DPX+SNY*DPY)*(SNX*SIY-SNY*SIX)
95 DVB=(SNPX*SIX+SNX*SIPX+SNPY*SIY+SNY*SIPY)*
96 2(SNX*DY-SNY*DX)
97 DVB=DVB+(SNPX*SIY+SNX*SIPY-SNPY*SIX-SIY*SIPX)*
98 2(SNX*DX+SNY*DY)*
99 DVB=DVB+(SNX*SIX+SNY*SIY)*(SNPX*DY-SNPY*DX)
100 DVB=DVB+(SNPX*DX+SNFY*DY)*(SNX*SIY-SNY*SIX)
101 DV=-(FI+DVT*DPSR)/DVB
102 C!!! APPROXIMATE ANGLE VR FOR THE REFL POINT IN DIRECTION PHPR
103 VR=VR+DV
104 50 CONTINUE
105 C!!! CHECK ACCURACY OF REFLECTION POINT ANGLE CALCULATED
106 CSV=COS(VR)
107 SNV=SIN(VR)
108 SNX=B*CSV
109 SNY=A*SNV
110 DD=SQRT(SNX*SNX+SIY*SIY)
111 SNX=SNX/DD
112 SNY=SNY/DD
113 SIX=A*CSV-XIS(1)
114 SIY=B*SNV-XIS(2)
115 S=SQRT(SIX*SIX+SIY*SIY)
116 SIX=SIX/S
117 SIY=SIY/S
118 C!!! CALCULATE THE ERROR FUNCTION ERC, THE SUM OF DOT
119 C!!! PRODUCTS OF INCIDENT AND REFLECTED UNIT VECTORS AND
120 C!!! CYLINDER UNIT NORMAL (SHOULD BE CLOSE TO ZERO)
121 DS=SNX*CPX+SNY*SPX
122 DR=SNX*SIX+SNY*SIY
123 DOTP=.5*(DS-DR)
124 ERC=DS+DR
125 ERCA=ABS(ERC)
126 C!!! IF ERROR IS NOT SUFFICIENTLY SMALL, DOUBLE NUMBER OF STEPS
127 C!!! (UP TO 32) AND RECALCULATE VR
128 IF(ERCA.LT.0.0005) GO TO 80
129 C!!! IF MAX NUMBER OF STEPS ALREADY REACHED, PRINT WARNING
130 IF(IVD(MR).GE.32) GO TO 70
131 IVD(MR)=2*IVD(MR)
132 GO TO 40

```

```

133 90 CONTINUE
134 WRITE(6,1) ERC,VR,PHSR
135 1 FORMAT(' ERROR IN REPTCL= ',3F12.6)
136 80 CONTINUE
137 C!!! IF ERROR IS VERY SMALL, DIVIDE NUMBER OF ITERATION
138 C!!! STEPS USED IN HALF FOR NEXT TIME ROUTINE IS CALLED
139 IF(ERCA.GE.0.000005) GO TO 90
140 IF(IVD(MR).EQ.1) GO TO 90
141 IVD(MR)=IVD(MR)/2
142 90 CONTINUE
143 VRO(MR)=VR
144 PHOR(MR)=PHSR
145 PHORP(MR)=PHSPR
146 IF(.NOT.LRFC) IVD(MR)=1
147 LRFC=.TRUE.
148 RETURN
149 END

```

ROTRAN

PURPOSE

To transform a point or vector defined in the old reference coordinate system to the new (cylinder-centered) reference coordinate system representation. This is used in the main program to perform the reference coordinate system transformation.

PERTINENT GEOMETRY

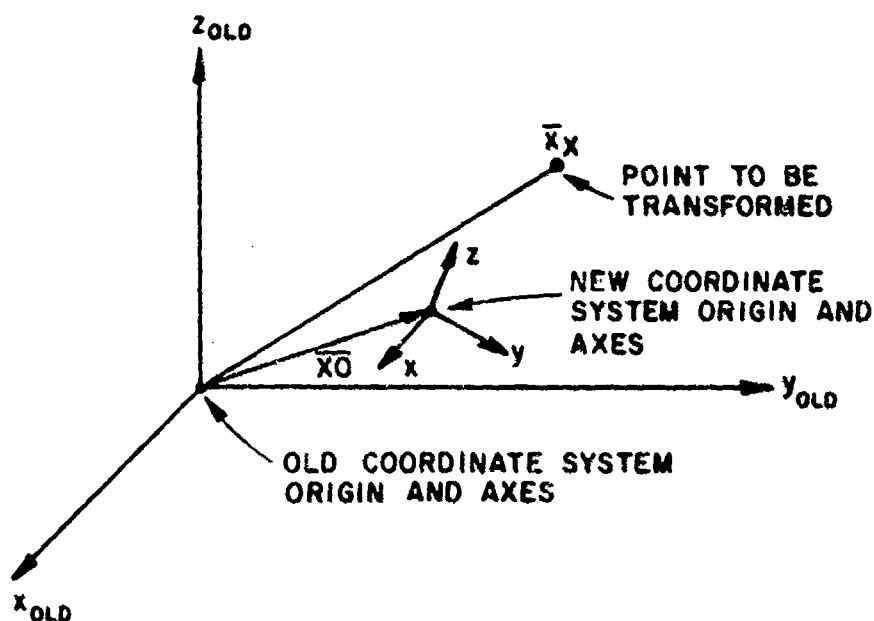


Figure 97-- Illustration of old and new reference coordinate systems.

$$\bar{x}_x = \hat{x}_{old} XX(1) + \hat{y}_{old} XX(2) + \hat{z}_{old} XX(3)$$

$$\bar{x}_x = \hat{x} XRT(1) + \hat{y} XRT(2) + \hat{z} XRT(3)$$

METHOD

The point $\bar{x}_x$ defined in the old coordinate system may be represented by point $\bar{x}_{rt}$ in the new coordinate system where:

$$\bar{X}_{rt} = \begin{bmatrix} v_{c1} \end{bmatrix} \bar{X}_t, \text{ where } \bar{X}_t = \bar{X}_x - \bar{X}_0$$

or

$$\begin{bmatrix} XRT(1) \\ XRT(2) \\ XRT(3) \end{bmatrix} = \begin{bmatrix} XCL(1) & XCL(2) & XCL(3) \\ YCL(1) & YCL(2) & YCL(3) \\ ZCL(1) & ZCL(2) & ZCL(3) \end{bmatrix} \begin{bmatrix} XX(1) - XO(1) \\ XX(2) - XO(2) \\ XX(3) - XO(3) \end{bmatrix} .$$

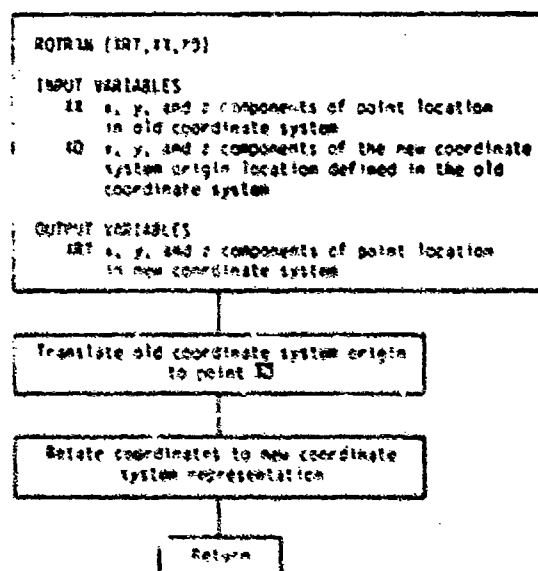
where $\bar{X}$ is the location of the new coordinate system origin defined in the old coordinate system and $\hat{x}$, $\hat{y}$, $\hat{z}$ are unit vectors defining the new coordinate system axes in old coordinate system components:

$$\hat{x} = \hat{x}_{old} XCL(1) + \hat{y}_{old} XCL(2) + \hat{z}_{old} XCL(3)$$

$$\hat{y} = \hat{x}_{old} YCL(1) + \hat{y}_{old} YCL(2) + \hat{z}_{old} YCL(3)$$

$$\hat{z} = \hat{x}_{old} ZCL(1) + \hat{y}_{old} ZCL(2) + \hat{z}_{old} ZCL(3).$$

FLOW DIAGRAM



SYMBOL DICTIONARY

XT X,Y, AND Z COMPONENTS OF POINT LOCATION AFTER TRANSLATING
OLD COORDINATE SYSTEM ORIGIN TO POINT XO

CODE LISTING

```

1 SUBROUTINE ROTRAN(XT,XX,XO)
2 C!!!
3 C!!! COORDINATE TRANSLATION AND ROTATION: XO IS THE
4 C!!! NEW ORIGIN: XCL,YCL,ZCL DEFINE THE NEW AXES.
5 C!!!
6 C!!!
7 DIMENSION XHT(3),XX(3),XO(3),XT(3)
8 LOGICAL LDEBUG,LTEST
9 COMMON/ROTRDI/XCL(3),YCL(3),ZCL(3)
10 COMMON/TEST/LDEBUG,LTEST
11 C!!! TRANSLATION OF COORDINATES
12 DO 10 N=1,3
13 XT(N)=XHT(N)-XO(N)
14 C!!! ROTATION OF COORDINATES
15 XHT(1)=XT(1)*XCL(1)+XT(2)*XCL(2)+XT(3)*XCL(3)
16 YHT(2)=XT(1)*YCL(1)+XT(2)*YCL(2)+XT(3)*YCL(3)
17 ZHT(3)=XT(1)*ZCL(1)+XT(2)*ZCL(2)+XT(3)*ZCL(3)
18 IF(.NOT.LTEST) RETURN
19 WRITE(6,500)
20 500 FORMAT(7,' TESTING ROTRAN SUBROUTINE')
21 WRITE(6,*) IHT
22 WRITE(6,*) JI
23 WRITE(6,*) JO
24 RETURN
25 END

```

RPLDPL

PURPOSE

To calculate the far-zone electric field (with phase referred to the RCS origin) for a source ray that reflects off plate MR and is then diffracted off edge ME of plate MP.

PERTINENT GEOMETRY

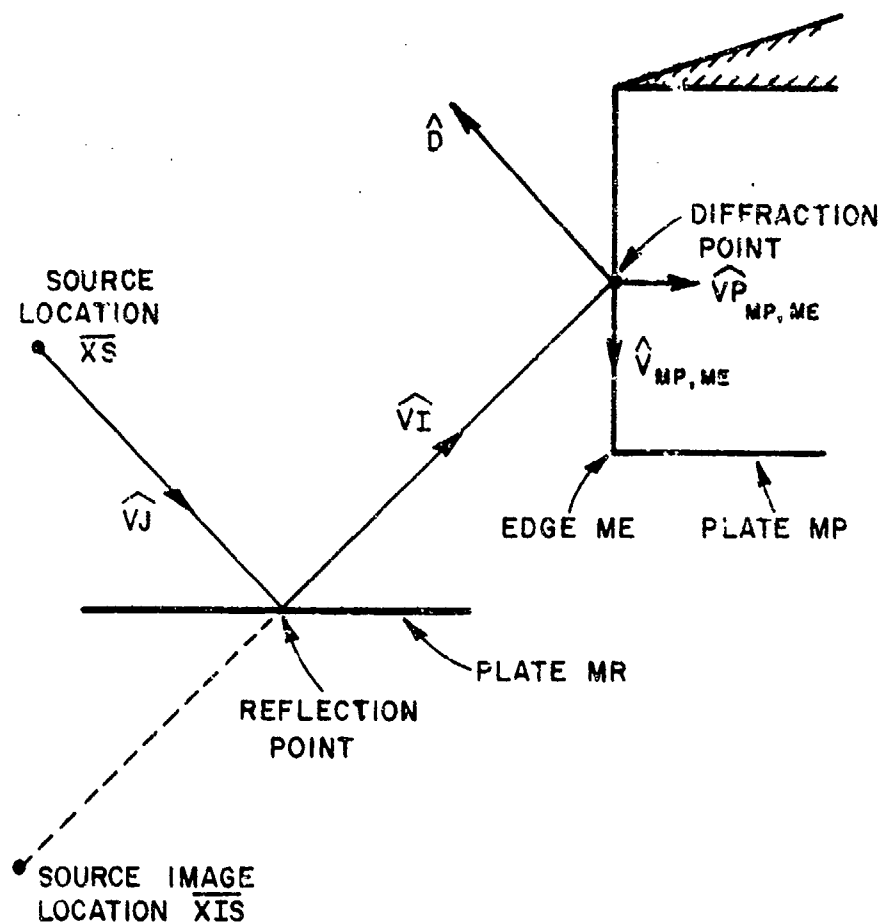


Figure 98--Illustration of a ray reflected by a plate and then diffracted by a plate edge.

METHOD

The field reflected by a plate and then diffracted by another plate edge is calculated in this subroutine [4,9,10]. The field reflected from the plate is found using image theory. The diffracted and slope diffracted fields of the plate edges and corners are obtained as described in subroutine DIFPLT. The diffracted edge and slope fields are combined and the phase is referred to the reference

coordinate system origin by the factor $e^{jk\hat{D}\cdot\overline{XDP}}$. The form of the field is therefore given by

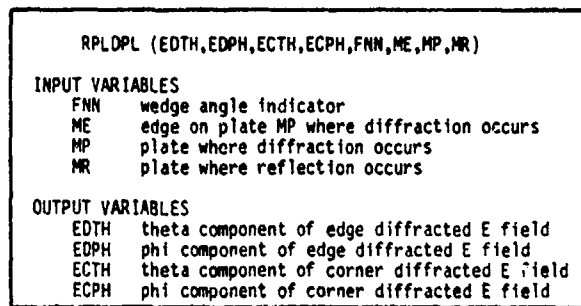
$$E^d = W_m(EDTH\hat{\theta} + EDPH\hat{\phi}) \frac{e^{-jkR}}{R}.$$

The corner and slope corner diffracted fields are combined in a similar way and are given by

$$E^c = W_m(ECTH\hat{\theta} + ECPH\hat{\phi}) \frac{e^{-jkR}}{R}$$

where the factor $\frac{e^{-jkR}}{R}$ and the source (W_m) weight are added elsewhere in the code.

FLOW DIAGRAM



1. Specify single reflection source image location
2. Perform diffraction point geometry calculations

Determine permissible range
for diffraction angle

Determine
if diffraction
exists

Yes

Compute edge diffraction point
and incident ray unit vector $\hat{v}_i$

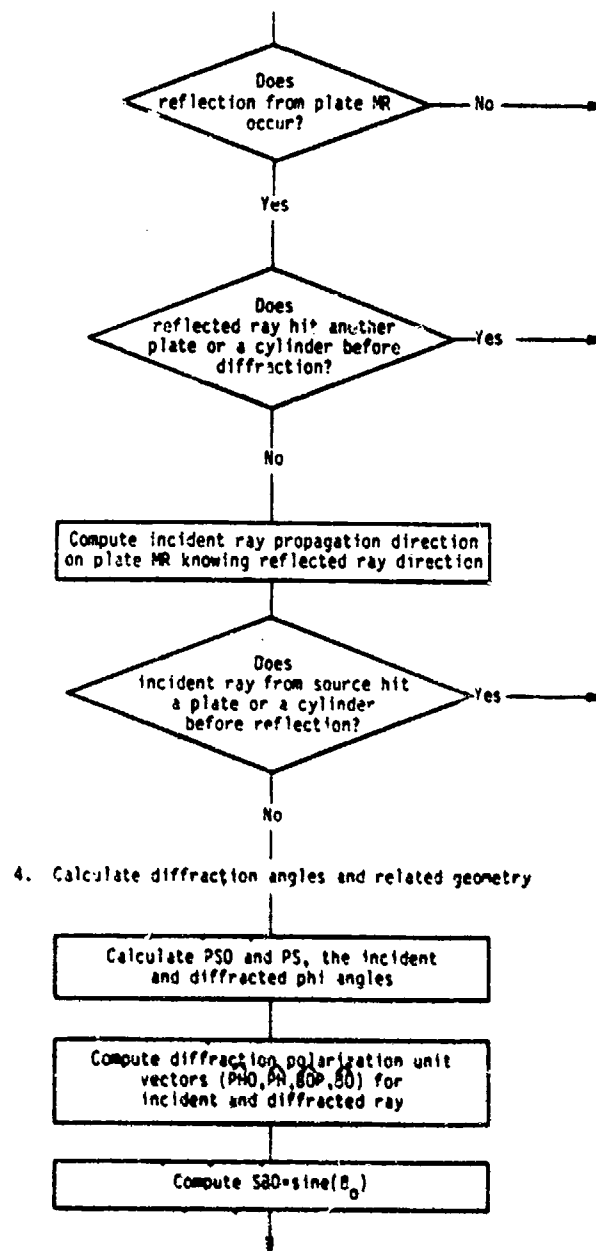
Is diffraction point on edge ME?
If not, set at appropriate
corner and set LOIF false

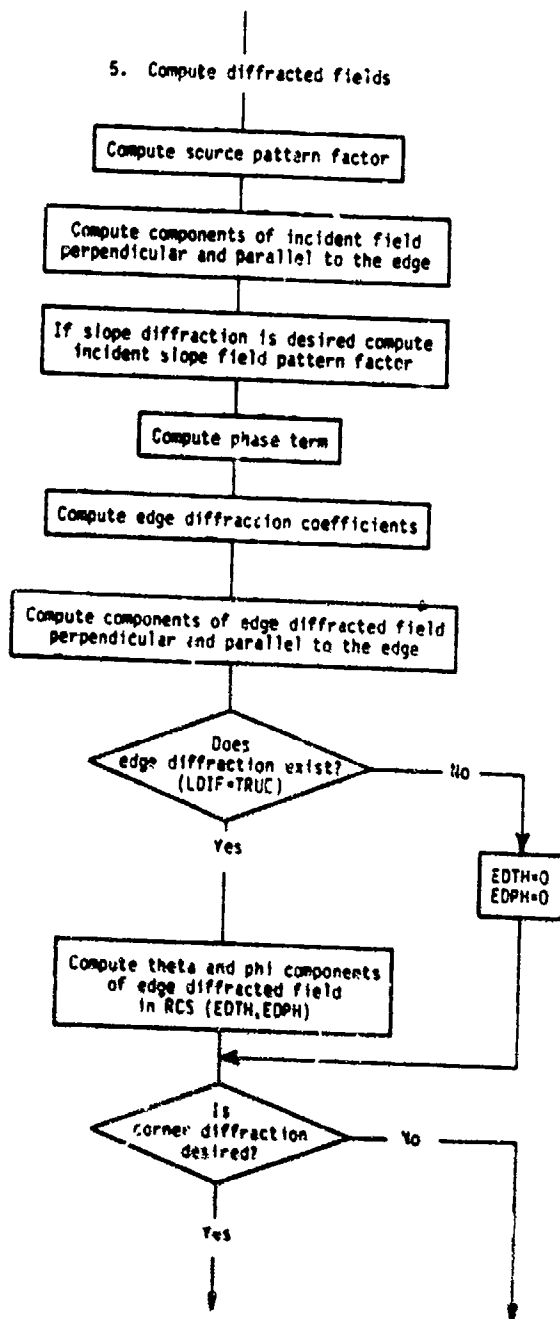
3. Check to see if ray is shadowed

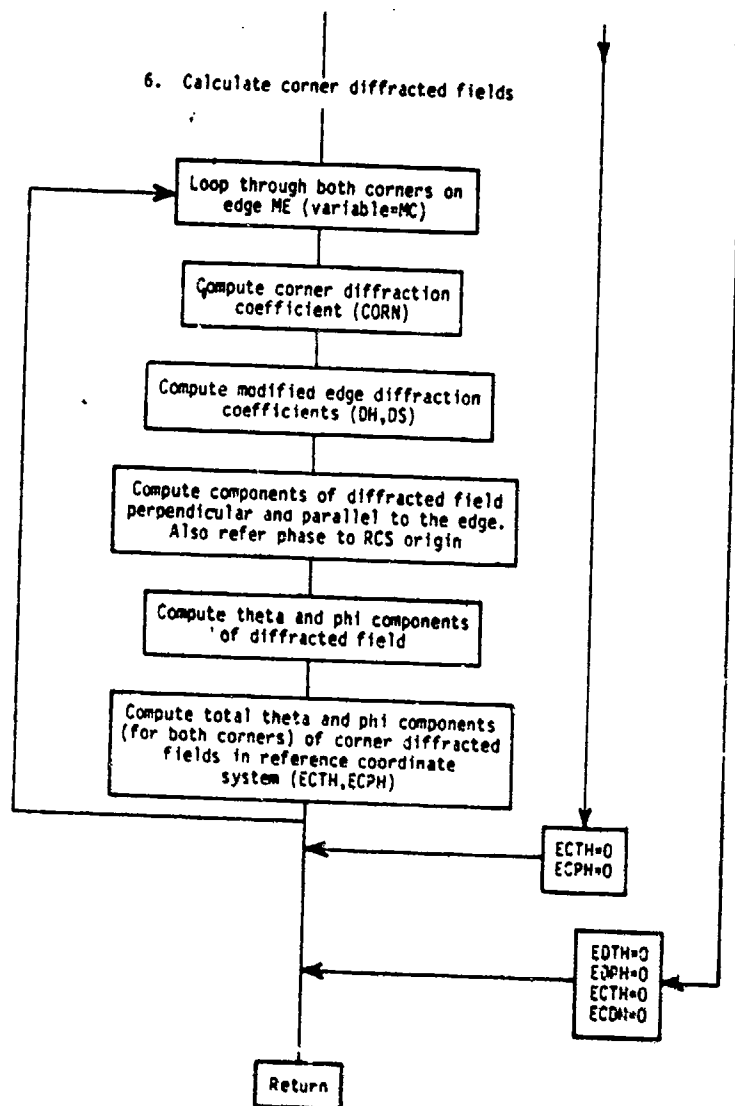
Does
diffracted ray hit another
plate or a cylinder?

No

Yes







SYMBOL DICTIONARY

ADN	DOT PRODUCT OF VECTOR FROM PLATE MP TO THE SOURCE IMAGE AND THE PLATE UNIT NORMAL
AFN	WEDGE ANGLE NUMBER
BDEL	VARIABLE USED TO EXPAND DIFFRACTION ANGLE RANGE IF CORNER DIFFRACTION IS USED
BUHI	UPPER LIMIT FOR ED, THE COSINE OF THE DIFFRACTION ANGLE BETA
BDLCW	LOWER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA
BETN	DIFFERENCE IN DIFFRACTED AND INCIDENT PHI ANGLES
BETP	SUM OF DIFFRACTED AND INCIDENT PHI ANGLES
BO	DIFFRACTED FIELD BETA POLARIZATION UNIT VECTOR IN DIFFRACTION EDGE COORDINATE SYSTEM (IN X,Y,Z RCS COMPONENTS)
BOP	INCIDENT FIELD BETA POLARIZATION UNIT VECTOR IN DIFFRACTION EDGE COORDINATE SYSTEM (IN X,Y,Z RCS COMPONENTS)
BRD	LOWER AND UPPER LIMIT FOR EDGE DIFFRACTION ANGLE BRD(1)=COS(ELON) BRD(2)=COS(EHIGH)
CNP	COSINE OF HALF WEDGE ANGLE
CORN	CORNER DIFFRACTION COEFFICIENT
CPH	COSINE OF PSR
CPHJ	COSINE OF PHJR
CPHC	COSINE OF PSOR
CTH	COSINE OF THR
CTHJ	COSINE OF THJR
CTHP	COSINE OF THPR
DEL	PARAMETER USED IN TRANSITION FUNCTION
DH	DIFFRACTION COEF. FOR HARD BOUNDARY CONDITION
DHIR	DISTANCE FROM REFLECTION POINT TO DIFFRACTION POINT
DHI1	DISTANCE FROM SOURCE TO REFLECTION POINT (FROM PLAIN)
DHI	DISTANCE FROM SOURCE TO HIT (FROM PLAIN AND CYLIND)
DIN	EDGE DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR INCIDENT DIFFRACTED FIELD
DIP	EDGE DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR REFLECTED DIFFRACTED FIELD
DPH	SLOPE DIFFRACTION COEFFICIENT FOR HARD BOUNDARY CONDITION
DPS	SLOPE DIFFRACTION COEFFICIENT FOR SOFT BOUNDARY CONDITION
DS	DIFFRACTION COEF. FOR SOFT BOUNDARY CONDITION
DV	DOT PRODUCT OF EDGE UNIT VECTOR AND DIFFRACTED RAY PROPAGATION DIRECTION UNIT VECTOR
ECPH	PHI COMPONENT OF CORNER DIFFRACTED E-FIELD
ECTH	THETA COMPONENT OF CORNER DIFFRACTED E-FIELD
EDPH	PHI COMPONENT OF EDGE DIFFRACTED E-FIELD
EDPL	COMPONENT OF DIFFRACTED FIELD PARALLEL TO THE EDGE
EDPH	COMPONENT OF DIFFRACTED FIELD PERPENDICULAR TO THE EDGE
EDTH	THETA COMPONENT OF EDGE DIFFRACTED E-FIELD
EP	THETA COMPONENT OF CORNER DIFFRACTED E-FIELD IN RCS
EG	PHI COMPONENT OF CORNER DIFFRACTED E-FIELD IN RCS
EIPL	COMPONENT OF INCIDENT FIELD PARALLEL TO THE EDGE
EIPLP	PATTERN FACTOR FOR COMPONENT OF INCIDENT SLOPE FIELD PARALLEL TO THE EDGE
EIPK	COMPONENT OF INCIDENT FIELD PERPENDICULAR TO THE EDGE
EIPKP	PATTERN FACTOR FOR COMPONENT OF INCIDENT SLOPE FIELD PERPENDICULAR TO THE EDGE
EIX	
EIY	SOURCE PATTERN FACTORS FOR X,Y, AND Z COMPONENTS OF INCIDENT E FIELD
EIZ	
EXPH	COMPLEX PHASE TERM (REFER PHASE TO RCS, ORIGIN)
FN	WEDGE ANGLE NUMBER
FNN	WEDGE ANGLE INDICATOR
FND	ANGLE EXTERIOR TO WEDGE IN DEGREES
GAK	DOT PRODUCT OF THE DIF RAY DIRECTION AND THE VECTOR FROM THE REF COORD SYS ORIGIN TO THE DIFFRACTION POINT
ISN	SIGN CHANGE VARIABLE
J	INDEX VARIABLE
LHIT	SET TRUE IF RAY HITS A PLATE OR CYLINDER (FROM PLAIN OR CYLIND)
NC	INDEX VARIABLE USED TO STEP THRU CORNERS

ME EDGE ON PLATE MP WHERE DIFFRACTION OCCURS
 MEC CORNER AT END OF EDGE ME
 MP PLATE FOR WHICH DIFFRACTION OCCURS
 MR PLATE WHERE REFLECTION OCCURS
 N DO LOOP VARIABLE
 NI DO LOOP VARIABLE
 NJ DO LOOP VARIABLE
 PD DOT PRODUCT OF DIF EDGE BINORMAL AND DIF RAY PROPAGATION DIRECTION
 PH DIFFRACTED FIELD PHI POLARIZATION UNIT VECTOR IN DIFFRACTION EDGE-FIXED COORDINATE SYSTEM (IN X,Y,Z RCS COMPONENTS)
 PHIR PHI COMPONENT OF REFL RAY PROPAGATION DIRECTION IN REF COORD SYS.
 PHJR PHI COMPONENT OF INCIDENT (SOURCE) RAY PROPAGATION DIRECTION
 PHC INCIDENT FIELD PHI POLARIZATION UNIT VECTOR IN DIFFRACTION EDGE-FIXED COORDINATE SYSTEM (IN X,Y,Z RCS COMPONENTS)
 PHSK PHI COMPONENT OF RAY PROPAGATION DIRECTION AFTER DIFFRACTION IN RCS
 PP NEGATIVE DOT PRODUCT OF DIF EDGE BINORMAL AND INCIDENT RAY UNIT VECTOR
 PS PSR*DPR
 PSD DIFFRACTED RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM
 PSO PSOR*DPR
 PSDC INCIDENT RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM
 PSCR PHI COMPONENT OF INCIDENT RAY DIRECTION IN EDGE FIXED COORDINATE SYSTEM
 PSK PHI COMPONENT OF DIFFRACTED RAY PROPAGATION DIRECTION IN EDGE-FIXED COORDINATE SYSTEM
 OD DOT PRODUCT OF DIF PLATE NORMAL AND DIF RAY PROPAGATION DIRECTION
 OI NEGATIVE OF DOT PRODUCT OF DIF PLATE NORMAL AND INCIDENT RAY PROPAGATION DIRECTION
 SBC SINE OF θ_0 , THE ANGLE THE DIFFRACTED RAY MAKES WITH THE EDGE
 SNP SINE OF HALF WEDGE ANGLE
 SP DISTANCE FROM SOURCE IMAGE TO DIFFRACTION POINT (FROM SUB. DFPTND)
 SPH SINE OF PSR
 SPHJ SINE OF PHJR
 SPHO SINE OF PSOR
 TDP DISTANCE FROM SOURCE IMAGE TO MODIFIED DIFFRACTION POINT
 THJ SINE OF THJR
 THK SINE OF THR
 TEK COEFFICIENT OF CORNER DIFFRACTED FIELDS
 THIA THETA COMPONENT OF REFLECTED RAY DIRECTION IN REF COORD SYS
 THJK THETA COMPONENT OF INCIDENT (SOURCE) RAY PROPAGATION DIRECTION
 THPR ANGLE DIFFRACTED RAY MAKES WITH EDGE
 THK ANGLE BETWEEN EDGE UNIT VECTOR AND RAY FROM SOURCE IMAGE LOCATION TO CORNER MC
 TDP DISTANCE PARAMETER USED IN CALCULATING DIFFRACTION COEFFICIENTS
 VAX 3X3 MATRIX DEFINING THE SOURCE IMAGE COORD SYS. AXES
 VC UNIT VECTOR FROM SOURCE IMAGE TO CORNER 1 OR 2 OF EDGE ME
 VCM DISTANCE FROM SOURCE IMAGE TO CORNER 1 OR 2 OF EDGE ME
 VECT VECTOR USED TO MOVE DIFFRACTION POINT OFF EDGE FOR SHADOWING TESTS
 VI UNIT VECTOR OF RAY INCIDENT ON EDGE FROM PLATE REFLECTION (FROM SUB. DFPTND)
 VIP UNIT VECTOR OF RAY FROM SOURCE IMAGE TO MODIFIED DIF POINT
 VJ X,Y, AND Z COMPONENTS OF SOURCE RAY PROPAGATION DIRECTION
 VMC DISTANCE ALONG THE EDGE FROM FIRST CORNER OF EDGE TO DIFFRACTION POINT
 XD DIFFRACTION POINT (CALCULATED IN SUB. DFPTND) IN RCS
 XDP MODIFIED DIFFRACTION POINT USED FOR SHADOWING TESTS
 XIC SOURCE IMAGE LOCATION (FOR REFLECTION FROM PLATE MR)
 XS SOURCE LOCATION IN REF COORD SYS
 ZP DOT PRODUCT OF PROPAGATION DIRECTION UNIT VECTOR AND VECTOR FROM DIFFRACTION POINT TO CORNER MC

CODE LISTING

```

1 C-----
2      SUBROUTINE RPLDPL(EDTH,EDPH,ECTH,ECPH,FNN,ME,MP,MR)
3 C!!!
4 C!!! DETERMINES THE REFLECTED/DIFFRACTED FIELD WITH PHASE
5 C!!! REFERRED TO ORIGIN. RAY IS REFLECTED FROM PLATE #MR AND
6 C!!! DIFFRACTED FROM EDGE #ME ON PLATE #MP.
7 C!!!
8      COMPLEX EF,EG,EIPR,EIPL,EXPH,DIN,DIP,EDPR,EDPL,EDTH,EDPH
9      COMPLEX EIPRP,EIPLP,EIX,EIY,EIZ,CORN,FFCT
10     COMPLEX DH,DS,DPH,DPS,ECBI,ECBR,ECTH,ECPH
11     DIMENSION VI(3),XD(3),PHO(3),PH(3),BOP(3),BO(3),XDP(3)
12     DIMENSION XIS(3),VJ(3),VC(2,3),VCM(2),BRD(2),VT(3),VIP(3)
13     DIMENSION VAX(3,3)
14     LOGICAL LHIT,LSURF
15     LOGICAL LDEBUG,LTEST,LSLOPE,LCORNR,LDIF
16     COMMON/TEST/LDEBUG,LTEST
17     COMMON/LOGDIF/LSLOPE,LCORNR
18     COMMON/EDMAG/VMAG(14,6)
19     COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
20     2,MEP(14),XPX
21     COMMON/SORINF/XS(3),VXS(3,3)
22     COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
23     COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
24     COMMON/1HPHUV/DI(3),DP(2)
25     COMMON/PIS/PI,TPI,OPR,RPO
26     COMMON/SURFAC/LSURF(14)
27     FN=FNN
28 C!!! INITIALIZE FIELDS
29     EDTH=(0.,0.)
30     EDPH=(0.,0.)
31     ECTH=(0.,0.)
32     ECPH=(0.,0.)
33     IF (LDEBUG) WRITE (6,106)
34 106  FORMAT (/,' DEBUGGING RPLDPL SUBROUTINE')
35     MEC=ME+1
36     IF(MEC.GT.MEP(MP)) MEC=1
37     DV=0.
38     DO 10 N=1,3
39     DV=DV+DIN)*V(MP,ME,N)
40     IF(ABS(DV).GT.0.999) GO TO 40
41 C!!! 1. SPECIFY SINGLE REFLECTION SOURCE IMAGE LOCATION
42 10   XIS(N)=XI(MR,MR,N)
43 C!!! 2. PERFORM DIFFRACTION POINT GEOMETRY CALCULATIONS
44 C!!! DETERMINE PERMISSABLE RANGE FOR DIFFRACTION ANGLE
45     VCM(1)=0.
46     VCM(2)=0.
47     BRD(1)=0.
48     BRD(2)=0.
49     DO 11 N=1,3
50     VC(1,N)=X(MP,ME,N)-X(MR,MR,N)
51     VC(2,N)=X(MP,ME,N)-X(MR,MR,N)
52     VCM(1)=VCM(1)+VC(1,N)*VC(1,N)
53 11   VCM(2)=VCM(2)+VC(2,N)*VC(2,N)
54     VCM(1)=SQRT(VCM(1))
55     VCM(2)=SQRT(VCM(2))
56     DO 12 J=1,2
57     DO 12 N=1,3
58     VC(J,N)=VC(J,N)/VCM(J)
59 12   BRD(J)=BRD(J)+V(VP,ME,N)*VC(J,N)
60     BDEL=0.
61     IF (LCORNR) BDEL=1.3
62     BLOW=BRD(1)-BDEL
63     BHI=BRD(2)+BDEL
64 C!!! DETERMINE IF DIFFRACTION EXISTS
65     IF(DV.(1.BLOW.OR.DV.GT.BHI)) GO TO 4*
66 C!!! COMPUTE FINAL DIFFRACTION POINT AND THE RAY UNIT VECTOR VI

```

```

67 CALL DFPTWD(XIS,DV,VI,SP,XD,ME,MP)
68 VMG=0.
69 ADN=0.
70 AFN=FNI
71 IF (AFN.GT.2.)AFN=6.-AFN
72 CNP=COS(AFN*PI/2.)
73 SNP=SIN(AFN*PI/2.)
74 DO 15 N=1,3
75 XDP(N)=XD(N)
76 VMG=VMG+(XD(N)-X(MP,ME,N))*V(MP,ME,N)
77 15 ADN=ADN+(XI(MR,MR,N)-X(MP,1,N))*VN(MP,N)
78 LDIF=.TRUE.
79 C!!! IS DIF POINT ON EDGE ME?
80 C!!! IF NOT, SET AT APPROPRIATE CORNER AND SET LDIF FALSE
81 IF (VMG.LT.0.) GO TO 101
82 IF (VMG.LT.VMAG(MP,ME)-1.E-4) GO TO 102
83 DO 103 N=1,3
84 103 XDP(N)=X(MP,ME,N)-1.E-4*V(MP,ME,N)
85 LDIF=.FALSE.
86 GO TO 102
87 101 DO 104 N=1,3
88 104 XDP(N)=X(MP,ME,N)+1.E-4*V(MP,ME,N)
89 LDIF=.FALSE.
90 102 DO 16 N=1,3
91 VECT=VP(MP,ME,N)*CNP+VN(MP,N)*SNP
92 16 XDP(N)=XDP(N)+1.E-5*VECT
93 C!!! 3. CHECK TO SEE IF RAY IS SHADOWED
94 C!!! DOES DIFFRACTED RAY HIT ANOTHER PLATE?
95 CALL PLAIN(XDP,D,DHT,MP,LHIT)
96 IF (LHIT) GO TO 40
97 C!!! DOES DIFFRACTED RAY HIT A CYLINDER?
98 CALL CYLINT(XDP,D,PHSR,DHT,LHIT,.TRUE.)
99 IF (LHIT) GO TO 40
100 SPP=0.
101 DO 111 N=1,3
102 VIP(N)=XDP(N)-XIS(N)
103 111 SPP=SPP+VIP(N)*VIP(N)
104 SPP=SQRT(SPP)
105 DO 112 N=1,3
106 112 VIP(N)=VIP(N)/SPP
107 C!!! DOES REFLECTION FROM PLATE MR OCCUR?
108 CALL PLAIN(XIS,VIP,DHT,-MR,LHIT)
109 IF (.NOT.LHIT) GO TO 40
110 IF (DHT.GT.SPP) GO TO 40
111 DHR=SPP-DHT
112 DHR=DHR-1.E-3
113 C!!! DOES REFLECTED RAY HIT ANOTHER PLATE BEFORE DIFFRACTION?
114 CALL PLAIN(XIS,VIP,DHT,MR,LHIT)
115 IF (LHIT.AND.(DHT.LT.DHR)) GO TO 40
116 THIR=BTAN2(SQRT(VI(1)*VI(1)+VI(2)*VI(2)),VI(3))
117 PHIR=BTAN2(VI(2),VI(1))
118 C!!! DOES REFLECTED RAY HIT A CYLINDER.
119 CALL CYLINT(XIS,VI,PHIR,DHT,LHIT,.TRUE.)
120 IF (LHIT.AND.(DHT.LT.DHR)) GO TO 40
121 C!!! COMPUTE INCIDENT RAY DIRECTION ON PLATE MR
122 C!!! KNOWING REFLECTED DIRECT. AN.
123 CALL REFBP(PHJR,THJR,PHIR,THIR,MR)
124 SPU=SIN(PHJR)
125 CPU=COS(PHJR)
126 THU=SIN(THJR)
127 CTU=COS(THJR)
128 VJ(1)=SPU*STU
129 VJ(2)=SPU*CTU
130 VJ(3)=CTU
131 C!!! DOES INCIDENT RAY FROM SOURCE HIT A PLATE BEFORE REFLECTION?
132 CALL PLAIN(XS,VJ,DHT,MR,LHIT)

```

```

133      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 40
134 C!!! DOES INCIDENT RAY HIT A CYLINDER?
135      CALL CYLINT(XS,VJ,PHJR,DHT,LHIT,.FALSE.)
136      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 40
137 C!!! 4. CALCULATE DIF ANGLES AND RELATED GEOMETRY
138      QI=0.
139      PP=0.
140      OD=0.
141      PD=0.
142      DO 20 N=1,3
143      QI=QI-VN(MP,N)*VI(N)
144      PP=PP-VP(MP,ME,N)*VI(N)
145      OD=OD+VN(MP,N)*D(N)
146 20    PD=PD+VP(MP,ME,N)*D(N)
147 C!!! CALCULATE PSO AND PS, THE INCIDENT AND DIF PHI ANGLES
148      PSOR=BTAN2(QI,PP)
149      PSO=DPR*PSOR
150      IF(PSO.LT.0.) PSO=360.+PSO
151      PSR=BTAN2(OD,PD)
152      PS=DPR*PSR
153      IF(PS.LT.0.) PS=360.+PS
154      PSOD=PSO
155      PSD=PS
156      IF(FNN.LE.2.)GO TO 21
157      FN=FNN-2.
158      PSOD=360.-PSO
159      PSD=360.-PS
160 21    FNP=FN*180.+1.E-4
161      IF(PSO.GT.FNP.OR.PS.GT.FNP) GO TO 40
162      SPHO=SIN(PSOR)
163      CPHO=COS(PSOR)
164      SPH=SIN(PSR)
165      CPH=COS(PSR)
166 C!!! COMPUTE DIFFRACTION POLARIZATION UNIT
167 C!!! VECTORS (PHO,PH,BOP,BO)
168      DO 30 N=1,3
169      PHO(N)=-VP(MP,ME,N)*SPHO+.N(MP,N)*CPHO
170 30    PH(N)=-VP(MP,ME,N)*SPH+.N(MP,N)*CPH
171      BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
172      BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
173      BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
174      BO(1)=PH(2)*D(3)-PH(3)*D(2)
175      BO(2)=PH(3)*D(1)-PH(1)*D(3)
176      BO(3)=PH(1)*D(2)-PH(2)*D(1)
177 C!!! COMPUTE SBO=SINE(BO)
178      SBO=SQRT((V(MP,ME,3)*D(2)-V(MP,ME,2)*D(3))**2+(V(MP,ME,1)
179      2*D(3)-V(MP,ME,3)*D(1))**2+(V(MP,ME,2)*D(1)-V(MP,ME,1)*D(2))
180      2**2)
181      TFP=SP*SBO*SBO
182 C!!! 5. COMPUTE DIFFRACTED FIELDS
183 C!!! COMPUTE SOURCE PATTERN FACTORS
184      DO 20 NJ=1,3
185      DO 20 NI=1,3
186 20    VAX(NI,NJ)=VXT(NI,NJ,MPI)
187      CALL SOURCE(S,SG,SI,SIY,SI2,THIR,PHIR,VAX)
188 C!!! COMPUTE COMPONENTS OF INCIDENT FIELD PERP AND PARALLEL
189 C!!! TO THE EDGE
190      EIPR=E1*PHO(1)+E1Y*PHO(2)+E1Z*PHO(3)
191      E1PL=E1*BOP(1)+E1Y*BOP(2)+E1Z*BOP(3)
192 C!!! IF SLOPE DIF IS DESIRED, CALCULATE INCIDENT SLOPE FIELD
193 C!!! PATTERN FACTORS
194      IF(USLOPE)CALL SOURCE(FIPR,E1PL,VI,PHO,BOP,VAX)
195 C!!! COMPUTE PHASE TERM
196      CAR=ED(1)*D(1)+ED(2)*D(2)+ED(3)*D(3)
197      EIPH=CEPH(CMPL(2,THI)*(CAR-SP)/SQRT(SPI)
198 C!!! COMPUTE EDGE DIFFRACTION COEFFICIENTS

```

```

199 CALL DW (DS,DH,DPS,DPH,TPP,PSD,PSOD,SRC,FN,LSURF(MP))
200 IF (LDEBUG) WRITE (6,*) EIPR,EIPL
201 IF (LDEBUG) WRITE (6,*) DS,DH,DPS,DPH
202 IF (LDEBUG) WRITE (6,*) TPP,PSD,PSOD,SBO,FN
203 C!!! COMPUTE COMPONENTS OF EDGE DIF FIELD PERP. AND PARALLEL
204 C!!! TO THE EDGE
205 EDPR=-EIPR*DH*EXPR
206 EDPL=-EIPL*DS*EXPH
207 IF (.NOT.LSLOPE) GO TO 201
208 EDPR=EDPR-EIPR*DPH*EXPH/CNPLX(0.,TP1*SP*SBO)
209 EDPL=EDPL-EIPL*DPS*EXPH/CNPLX(0.,TP1*SP*SBO)
210 201 IF (.NOT.LDIF) GO TO 202
211 C!!! COMPUTE THETA AND PHI COMPONENTS OF EDGE DIF. FIELD
212 C!!! IF DIFFRACTION EXISTS
213 EDTH=EDPL*(BO(1)*DT(1)+BO(2)*DT(2)+BO(3)*DT(3))
214 2*EDPR*(PH(1)*DT(1)+PH(2)*DT(2)+PH(3)*DT(3))
215 EDPH=EDPL*(BO(1)*DP(1)+BO(2)*DP(2))
216 2*EDPR*(PH(1)*DP(1)+PH(2)*DP(2))
217 C!!! 6. IF CORNER DIF IS DESIRED, CALC CORNER DIF FIELDS
218 202 IF (.NOT.LCORN) GO TO 40
219 BETN=PSD-PSOD
220 BETP=PSD*PSOD
221 EF=(0.,0.)
222 EG=(0.,0.)
223 MC=ME-1
224 ISN=1
225 J=0
226 C!!! LOOP THRU BOTH CORNERS ON EDGE #ME
227 35 MC=MC+1
228 IF (MC.GT.MEP(MP)) MC=1
229 J=J+1
230 ISN=-ISN
231 CTH=-ISN*BRO(J)
232 CTHP=(ISN*OV
233 THPR=ACOS(CTHP)
234 THN=ACOS(CTH)
235 STHN=SIN(THN)
236 DEL=2.*TPI*VCN(J)*(COS(.5*(THN+THPR))+.2)
237 ZP=(X(MP,MC,1)-XD(1))*D(1)+(X(MP,MC,2)-XD(2))*D(2)
238 2*(X(MP,MC,3)-XD(3))*D(3)
239 TERM=-STHN/TPI/(CTH+CTHP)/SQRT(VCN(J))
240 C!!! COMPUTE CORNER DIFFRACTION COEFFICIENT (CORN)
241 CORN=TERM*FCT(DEL)*CEXP(CNPLX(0.,-TP1)*VCN(J)-SP-ZP)-.25*P(1)
242 CALL D1(DIN,TPP,BETN,SBO,FN,DEL,.TRUE.)
243 IF (LSURF(MP)) GO TO 311
244 CALL D1(DIP,TPP,BETP,SBO,FN,DEL,.TRUE.)
245 C!!! COMPUTE MODIFIED EDGE DIF. COEFFICIENTS (DH,DS)
246 DH=DIN*DIP
247 DS=DIN*DIP
248 GO TO 312
249 311 DH=DIN
250 DS=(0.,0.)
251 C!!! COMPUTE COMPONENTS OF DIF FIELD PERP. AND PARALLEL TO EDGE
252 312 EDPR=-EIPR*DH*EXPR
253 EDPL=-EIPL*DS*EXPH
254 IF (.NOT.LSLOPE) GO TO 201
255 EDPR=EDPR-EIPR*DPH*EXPH/CNPLX(0.,TP1*SP*SBO)
256 EDPL=EDPL-EIPL*DPS*EXPH/CNPLX(0.,TP1*SP*SBO)
257 C!!! COMPUTE THETA AND PHI COMPONENTS OF CORNER DIF FIELD
258 203 EDTH=EDPL*(BO(1)*DT(1)+BO(2)*DT(2)+BO(3)*DT(3))
259 2*EDPR*(PH(1)*DT(1)+PH(2)*DT(2)+PH(3)*DT(3))
260 EDPH=EDPL*(BO(1)*DP(1)+BO(2)*DP(2))
261 2*EDPR*(PH(1)*DP(1)+PH(2)*DP(2))
262 C!!! COMPUTE TOTAL THETA AND PHI COMPONENTS (FOR BOTH CORNERS)
263 C!!! OF CORNER DIF FIELDS IN REF COORD SYS.
264 EF=EF+EDTH+CORN

```

```

265      EG=EG+ECPH+CORN
266      IF (.NOT.LDEBUG) GO TO 36
267      WRITE (6,*) DS,DH,EDPR,EDPL
268      WRITE (6,*) ECTH,ECPH,CORN
269      WRITE (6,*) EF,EG
270 36    CONTINUE
271      IF (MC.EQ.ME) GO TO 35
272      ECTH=EF
273      ECPH=EG
274      RETURN
275 40    IF (.NOT.LTEST) GO TO 204
276      WRITE (6,205)
277 205    FORMAT (/,' TESTING RPLDPL SUBROUTINE')
278      WRITE (6,*) EDTH,EDPH,ECTH,ECPH
279      WRITE (6,*) FN,ME,MP,MR
280 204    RETURN
281      END

```

RPLRCL

PURPOSE

To compute the geometrical optics field reflected by a given plate and then reflected by the cylinder.

PERTINENT GEOMETRY

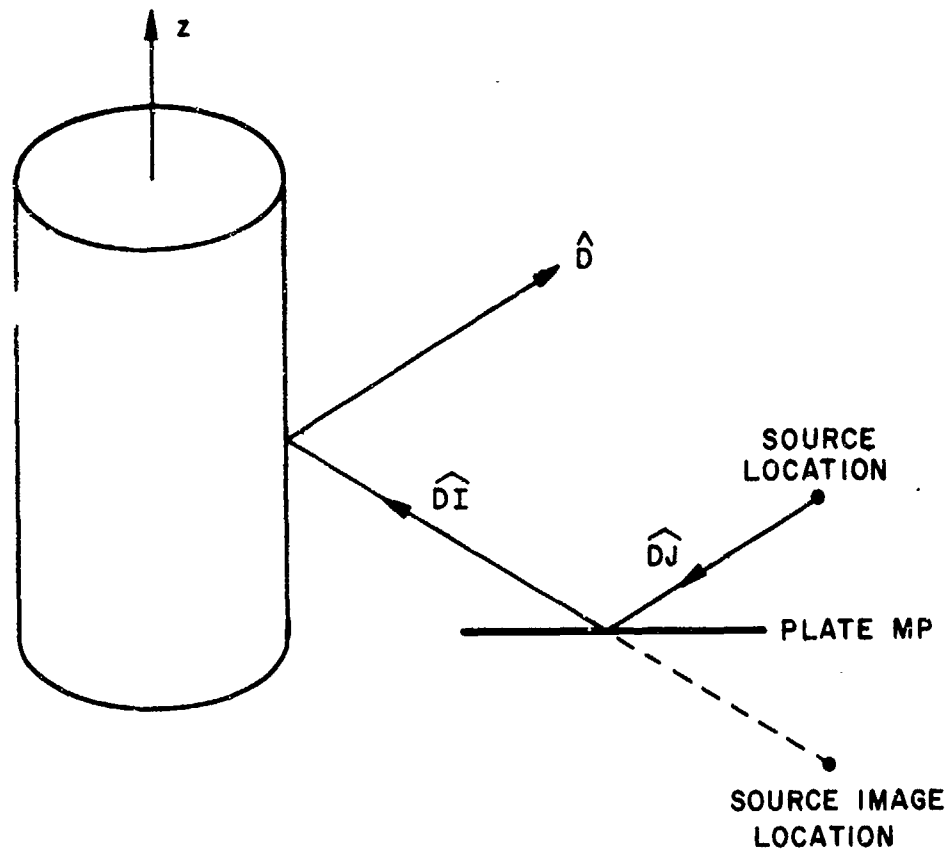


Figure 99--Illustration of plate reflected, cylinder reflected ray.

METHOD

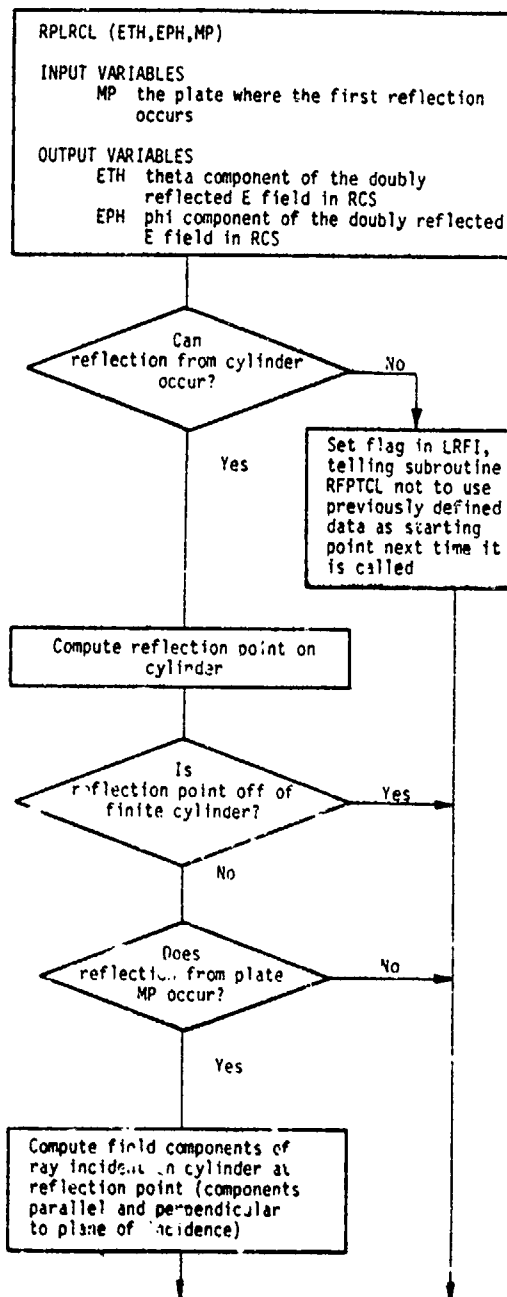
Subroutine RPLRCL functions as a service routine for subroutine RPLSCL, where the actual plate-cylinder fields are computed. The geometrical optics reflected field components ETH and EPH computed in RPLRCL are used only for reference purposes (when LOUT is set true). The field components calculated in RPLRCL which are used in RPLSCL, are the hard and soft components of the plate reflected field incident on the cylinder at the reflection point. These components, along with several other useful parameters, are passed to subroutine RPLSCL through common block FUDGI.

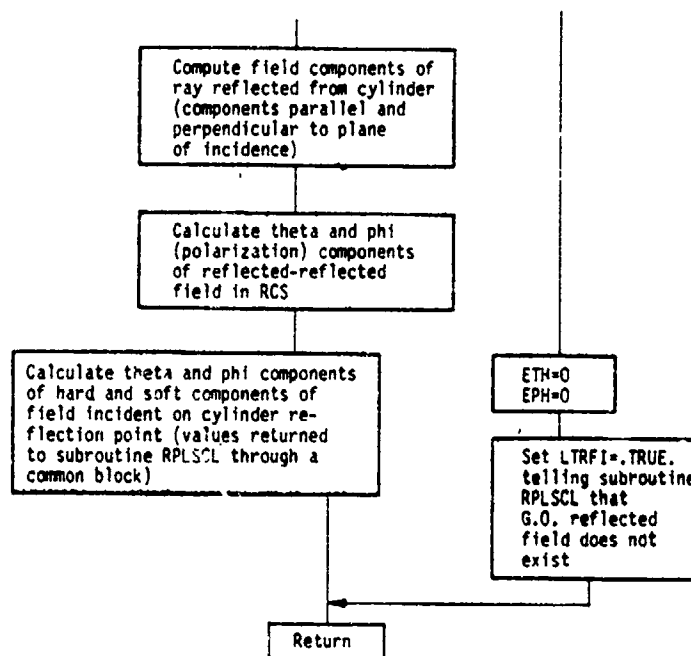
The geometrical optics fields determined in this subroutine for the reflection from the cylinder, are found in a similar manner to the fields calculated in subroutine REFCYL. However, in this subroutine the field incident on the cylinder is found from the image source for the particular plate of interest, as illustrated in Figure 99. The image source fields are calculated in a similar manner to those obtained in subroutine REFPLA. The phase of the resultant double reflected field is referred to the reference coordinate system origin. The double reflected field thus has the form

$$\vec{E}^{r,r} = W_m (ETH\hat{\theta} + EPH\hat{\phi}) \frac{e^{-jkR}}{R}$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM





SYMBOL DICTIONARY

CTHW DOT PRODUCT OF CYLINDER NORMAL AND REFLECTION
 PROPAGATION DIRECTION UNIT VECTOR
 CW COSINE OF ψ
 D PROPAGATION DIRECTION AFTER CYL REFL. IN (X,Y,Z) RCS COMPONENTS
 DD1 DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND
 CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 1 (2-D)
 DD2 DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND
 CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 2 (2-D)
 DHIS DISTANCE FROM REFLECTION POINT ON PLATE TO REFLECTION
 POINT ON THE CYLINDER
 DHIT DISTANCE FROM SOURCE TO HIT POINT (FROM PLAIN)
 DI X,Y, AND Z COMPONENTS OF INCIDENT RAY DIRECTION ON CYL IN RCS
 DJ X,Y,Z COMPONENTS OF PROPAGATION DIRECTION OF RAY INCIDENT
 ON PLATE
 DXY DOT PRODUCT OF VECTOR FROM ORIGIN TO SOURCE IMAGE
 LOCATION AND PROPAGATION DIRECTION (2-D)
 EF PATTERN FACTOR OF THETA COMPONENT OF INCIDENT FIELD IN RCS
 EG PATTERN FACTOR OF PHI COMPONENT OF INCIDENT FIELD IN RCS
 EHPH PHI COMPONENT OF THE HARD COMPONENT OF FIELD INCIDENT ON CYL
 (PARALLEL TO PLANE OF INCIDENCE)
 EHTH THETA COMPONENT OF THE HARD COMPONENT OF FIELD INCIDENT ON CYL
 (PARALLEL TO PLANE OF INCIDENCE)
 EIPP INCIDENT CYL FIELD COMPONENT PARALLEL TO PLANE OF INCIDENCE
 EIPR INCIDENT CYL FIELD COMPONENT PERPENDICULAR TO PLANE OF INC
 EPH PHI COMPONENT OF CYL REFLECTED E-FIELD
 ERPP CYL REFLECTED FIELD COMPONENT PARALLEL TO PLANE OF INCIDENCE
 ERPR CYL REFLECTED FIELD COMPONENT PERPENDICULAR TO PLANE OF INC.
 ERX } X,Y,Z COMPONENTS OF FIELD
 ERY } INCIDENT ON (OR REFLECTED FROM)
 ERZ } CYLINDER IN RCS
 ESPH PHI COMPONENT OF THE SOFT COMPONENT OF FIELD INCIDENT ON CYL
 (PERPENDICULAR TO PLANE OF INCIDENCE)
 ESTH THETA COMPONENT OF THE SOFT COMPONENT OF FIELD INCIDENT ON CYL
 (PERPENDICULAR TO PLANE OF INCIDENCE)
 ETH THETA COMPONENT OF CYL REFLECTED E FIELD
 EX }
 EY } PATTERN FACTOR OF X,Y,Z COMPONENTS OF SOURCE FIELD
 EZ } INCIDENT ON CYLINDER IN RCS
 LHIT SET TRUE IF RAY HITS PLATE (FROM PLAIN)
 LKFI SET TRUE IF REFL DATA IS AVAILABLE FROM PREVIOUS PATTERN
 ANGLE (OR FOR NEXT PATTERN ANGLE (WHEN LEAVING ROUTINE))
 LTRFI SET TRUE IF GEOMETRICAL OPTICS REFLECTED-REFLECTED
 FIELD DOES NOT EXIST
 PH COMPLEX PHASE AND RAY SPREADING COEFFICIENT
 PHIR PHI COMPONENT OF INCIDENT RAY DIRECTION ON CYL
 RH01 RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE
 AT REFLECTION POINT
 RH02 RAY SPREADING RADIUS NORMAL TO PLANE OF INCIDENCE
 AT REFLECTION POINT
 SHAG LENGTH OF RAY FROM REFL POINT ON CYL TO SOURCE IMAGE
 SUNH PART OF SPREADING FACTOR
 SXN }
 SYN } X,Y, AND Z COMPONENTS OF UNIT VECTOR OF RAY FROM REFL.
 SZN } POINT ON CYLINDER TO SOURCE IMAGE LOCATION IN RCS
 THIR THETA COMPONENT OF INCIDENT RAY DIRECTION ON CYL
 UIRPA }
 UIRPY } X,Y,Z COMPONENTS OF INCIDENT FIELD POLARIZATION UNIT VECTOR
 UIRPZ } PARALLEL TO PLANE OF INCIDENCE
 UIRPX }
 UIRPY } X,Y,Z COMPONENTS OF INC/REFL FIELD POLARIZATION UNIT VECTOR
 UIRPZ } PERPENDICULAR TO PLANE OF INCIDENCE
 UNIRPA }
 UNIRPY } X,Y,Z COMPONENTS OF REFL FIELD POLARIZATION UNIT VECTOR
 UNIRPZ } PARALLEL TO PLANE OF INCIDENCE
 VAX PATHS DEFINING SOURCE COORDINATE SYS AXES IN RCS COMPONENTS

AIS X,Y,Z COMPONENTS OF SOURCE IMAGE LOCATION
ALSO REFLECTION POINT ON PLATE
XR LOCATION OF REFLECTION POINT ON CYL IN (X,Y,Z) RCS

CODE LISTING

```

1 C-----
2 SUBROUTINE RPLRCL(ETH,EPH,MP)
3 C!!!
4 C!!! COMPUTES THE G.O. FIELD REFLECTED FROM PLATE #MP THEN
5 C!!! REFLECTED FROM THE ELLIPTIC CYLINDER
6 C!!!
7 DIMENSION UN(2),UB(2),DI(3),DJ(3),XIS(3),VAX(3,3)
8 COMPLEX ETH,EPH,EX,EY,EZ,PH,EIPR,EIPP,ERX,ERY,ERZ,ERPR,ERPP
9 COMPLEX ESTH,ESPH,EH,EHPH,TRAN,EF,E7
10 LOGICAL LHIT,LHFI,LDEBUG,LTEST,LTRFI
11 COMMON/FUDGI/TRAN,ESTH,ESPH,EH,EHPH,XR(3),RG,RHO1,SMAG,LTRFI
12 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
13 COMMON/SORINF/XS(3),VXS(3,3)
14 COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
15 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
16 2,MEP(14),MPX
17 COMMON/PIS/PI,TPI,DPR,RPD
18 COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
19 COMMON/THPHUV/DI(3),DP(2)
20 COMMON/ENDICL/DI(14),VTI(14,2),BTI(14,4)
21 COMMON/TEST/LDEBUG,LTEST
22 COMMON/CLRFI/LRFI(14)
23 IF(LDEBUG) WRITE(6,9999)
24 9999 FORMAT(/,' DEBUGGING RPLRCL SUBROUTINE')
25 LTRFI=.FALSE.
26 C!!! CAN REFLECTION FROM CYLINDER OCCUR?
27 IF(DTI(MP).LT.-1.5) GO TO 12
28 UXY=XI(MP,MP,1)*CPS+XI(MP,MP,2)*SPS
29 IF(UXY.GT.0.) GO TO 10
30 UD1=BTI(MP,1)*CPS+BTI(MP,2)*SPS
31 UD2=BTI(MP,3)*CPS+BTI(MP,4)*SPS
32 IF(DD1.GT.DTI(MP).AND.UD2.GT.DTI(MP)) GO TO 12
33 10 CONTINUE
34 C!!! CALCULATE REFLECTION POINT ON CYLINDER
35 CALL RPTCL(PHSR,MP,VN,DOTP,DD,S,LRFI(MP))
36 IF(DOTP.LE.0.) GO TO 11
37 XR(1)=A*COS(VR)
38 XR(2)=B*SIN(VR)
39 XR(3)=XI(MP,MP,3)+S*CTHS/STHS
40 C!!! IS REFLECTION POINT OFF OF FINITE CYLINDER?
41 IF(XR(3).GT.ZC(1)+XR(1)*CTC(1).OR.
42 2XR(3).LT.ZC(2)+XR(1)*CTC(2)) GO TO 11
43 C!!! DOES CYLINDER REFLECTED RAY HIT A PLATE?
44 CALL PLAIN(XR,D,DHT,2,LHIT)
45 IF(LHIT) GO TO 11
46 SXN=XI(MP,MP,1)-XR(1)
47 SYN=XI(MP,MP,2)-XR(2)
48 SZN=-3*CTHS/STHS
49 SMAG=SQRT(SXN*SXN+SYN*SYN+SZN*SZN)
50 SXN=SXN/SMAG
51 SYN=SYN/SMAG
52 SZN=SZN/SMAG
53 PHIR=ATAN2(-SYN,-SXN)
54 THIR=ATAN2(SQRT(SXN*SXN+SYN*SYN),-SZN)
55 SPHI=SIN(PHIR)
56 CPHI=COS(PHIR)
57 STHI=SIN(THIR)
58 CTHI=COS(THIR)
59 DI(1)=CPHI*STHI
60 DI(2)=SPHI*STHI
61 DI(3)=CTHI
62 DO 15 K=1,3
63 15 AIS(K)=XI(MP,MP,K)
64 C!!! DOES REFLECTION OFF OF PLATE MP OCCUR?
65 CALL PLAIN(XIS,DI,DHT,-MP,LHIT)
66 IF(.NOT.LHIT) GO TO 11

```

```

07      DHIS=SMAG-DHIT
08 C!!! DOES REFLECTED RAY HIT PLATE BEFORE THE CYLINDER?
09      CALL PLAIN(XS,DI,DHT,MP,LHIT)
10      IF(LHIT.AND.(DHT.LT.DHIS)) GO TO 11
11      CALL REFEB(PHJR,THJR,PHIR,THIR,MP)
12      SPHJ=SIN(PHJR)
13      CPHJ=COS(PHJR)
14      STHJ=SIN(THJR)
15      CTHJ=COS(THJR)
16      DJ(1)=CPHJ*STHJ
17      DJ(2)=SPHJ*STHJ
18      DJ(3)=CTHJ
19 C!!! DOES SOURCE RAY INC. ON PLATE MP HIT ANOTHER PLATE
20 C!!! OR THE CYLINDER FIRST?
21      CALL PLAIN(XS,DJ,DHT,MP,LHIT)
22      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 11
23      CALL CYLINT(XS,DJ,PHJR,DHT,LHIT,.FALSE.)
24      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 11
25      DO 20 NJ=1,3
26      DO 20 NI=1,3
27      VAX(NI,NJ)=VXI(NI,NJ,MP)
28 C!!! CALCULATE SOURCE PATTERN FACTOR
29      CALL SOURCE(EF,EG,EX,EY,EZ,THIR,PHIR,VAX)
30      IF(LDEB) WRITE(6,*) EF,EG
31      RG=DD*DD*DD/A/B
32      CALL NDNB(UN,UB,VR)
33      CTHN=UN(1)*D(1)+UN(2)*D(2)
34      NR=BTAN2(SXN*UB(1)+SYN*UB(2),SZN)
35      SW=SIN(NR)
36      CW=COS(NR)
37      SST2=SW*SW+CW*CW*CTHN*CTHN
38      RPQ2=SMAG
39      RHQ1=SMAG*RG*CTHN/(RG*CTHN+2.*SMAG*SST2)
100 C!!! COMPUTE POLARIZATION UNIT VECTORS
101 C!!! PERPENDICULAR AND PARALLEL TO PLANE OF INCIDENCE
102      UIPRX=SIN(NR-PI/2.)*UB(1)
103      UIPRY=SIN(NR-PI/2.)*UB(2)
104      UIPRZ=CCS(NR-PI/2.)
105      UIPPX=SYN*UIPRZ-SZN*UIPRY
106      UIPPY=SZN*UIPRX-SXN*UIPRZ
107      UIPPZ=SXN*UIPRY-SYN*UIPRX
108      URPPX=UIPRY*D(3)-UIPRZ*D(2)
109      URPPY=UIPRZ*D(1)-UIPRX*D(3)
110      URPPZ=UIPRX*D(2)-UIPRY*D(1)
111      PHI=CEXP(CMPLX(0.,-PI*(SMAG)))/SMAG
112 C!!! CALCULATE INCIDENT FIELD COMPONENTS PARALLEL
113 C!!! AND PERPENDICULAR TO PLANE OF INCIDENCE
114      EIPR=(UIPRX*EX+UIPRY*EY+UIPRZ*EZ)
115      EIPP=(UIPPX*EX+UIPPY*EY+UIPPZ*EZ)
116      PHI=PHI*CEXP(CMPLX(0.,PI*(XR(1)*D(1)+XR(2)*D(2)+XR(3)*D(3))))
117      SORH=SQRT(RHQ1*RHQ2)
118 C!!! COMPUTE REFLECTED FIELD COMPONENTS PARALLEL
119 C!!! AND PERPENDICULAR TO PLANE OF INCIDENCE
120      ERPH=-SORH*PHI*EIPR
121      ERPP=SORH*PHI*EIPP
122      TRAN=SORH*PHI
123      ERX=ERPH*UIPRX+ERPP*URPPX
124      ERY=ERPH*UIPRY+ERPP*URPPY
125      ERZ=ERPH*UIPRZ+ERPP*URPPZ
126 C!!! CALCULATE THETA AND PHI COMPONENTS OF REFLECTED-
127 C!!! REFLECTED FIELD
128      ETH=ERX*DT(1)+ERY*DT(2)+ERZ*DT(3)
129      EPH=ERX*DP(1)+ERY*DP(2)
130 C!!! COMPUTE THETA AND PHI COMPONENT OF SOFT COMPONENT OF
131 C!!! FIELD INC. ON CYLINDER
132      ERX=EIPH*UIPRX

```

```

133      ERY=EIPR*UIPRY
134      ERZ=EIPR*UIPRZ
135      ESTH=ERX*DT(1)+ERY*DT(2)+ERZ*DT(3)
136      ESPH=ERX*DP(1)+ERY*DP(2)
137 C!!!  COMPUTE THETA AND PHI COMPONENT OF HART COMPONENT OF
138 C!!!  FIELD IFC. OF CYLINDER
139      ERX=EIPP*URPPX
140      ERY=EIPP*URPPY
141      ERZ=EIPP*URPPZ
142      EHTH=ERX*DT(1)+ERY*DT(2)+ERZ*DT(3)
143      EHPH=ERX*DP(1)+ERY*DP(2)
144      GO TO 905
145 12    LRFI(MP)=.FALSE.
146 11    LTRFI=.TRUE.
147      ETH=(0.,0.)
148      EPH=(0.,0.)
149 905   CONTINUE
150      IF(.NOT.LTEST) RETURN
151      WRITE(6,910)
152 910   FORMAT(/,' TESTING KPLWCL SUBROUTINE')
153      WRITE(6,*) ETH,EPH,MP
154      RETURN
155      END

```

RPLRPL

PURPOSE

To calculate the far zone electric field due to double reflection from specified plates (reflection off of plate MP and then plate MPP).

PERTINENT GEOMETRY

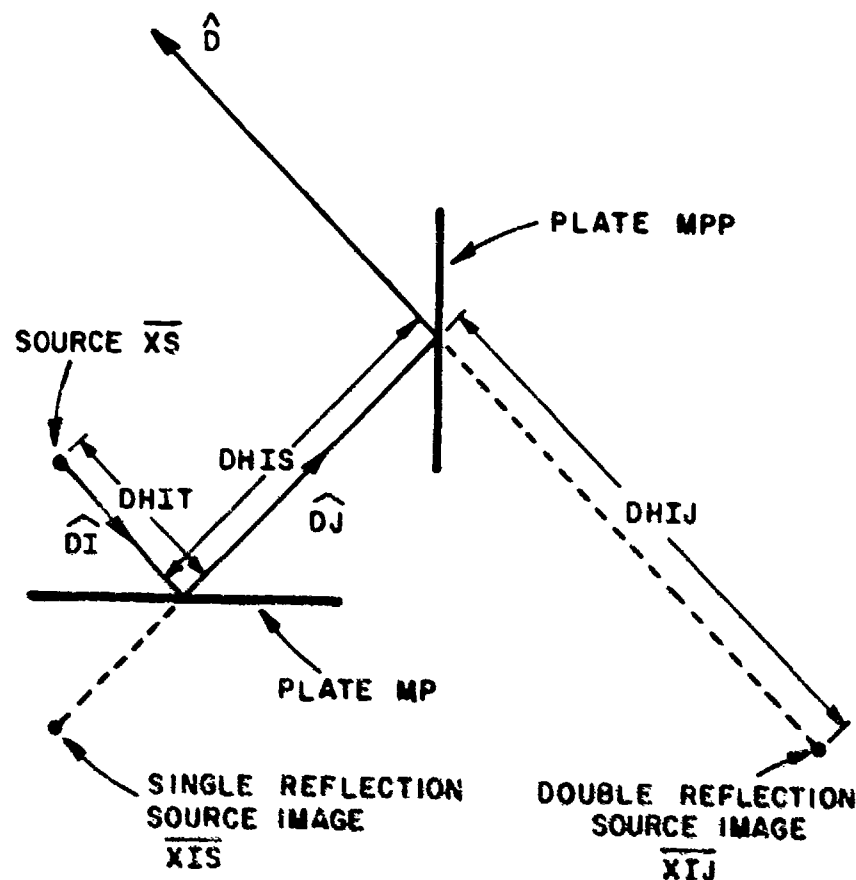


Figure 100--Geometry for double reflected ray.

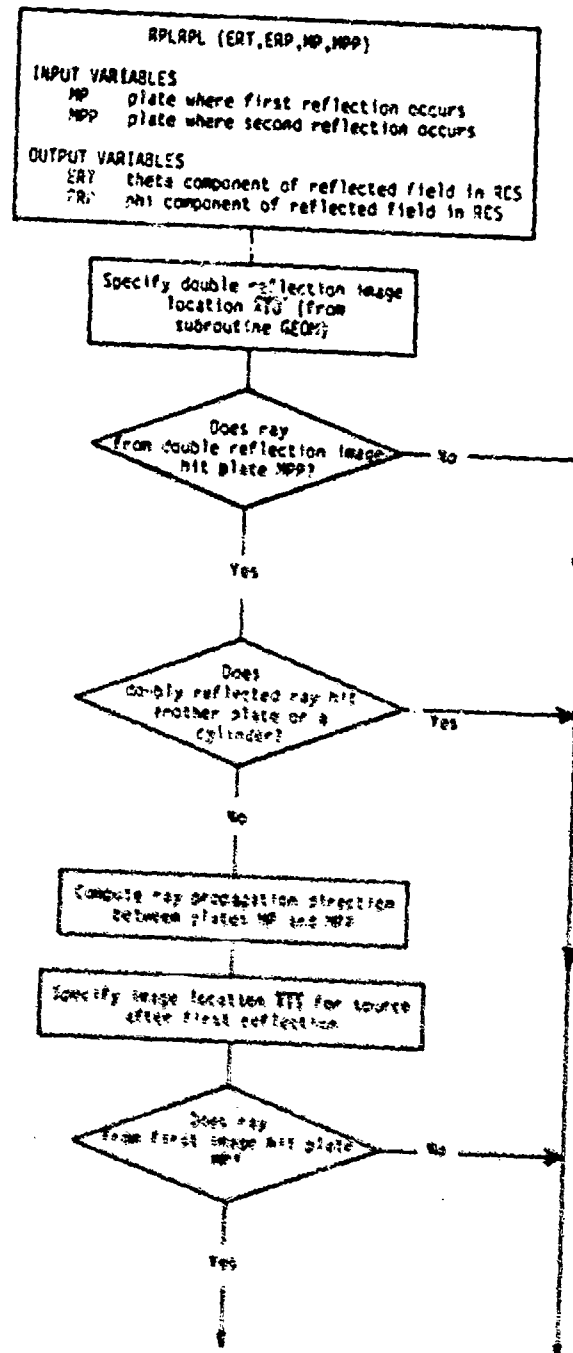
METHOD

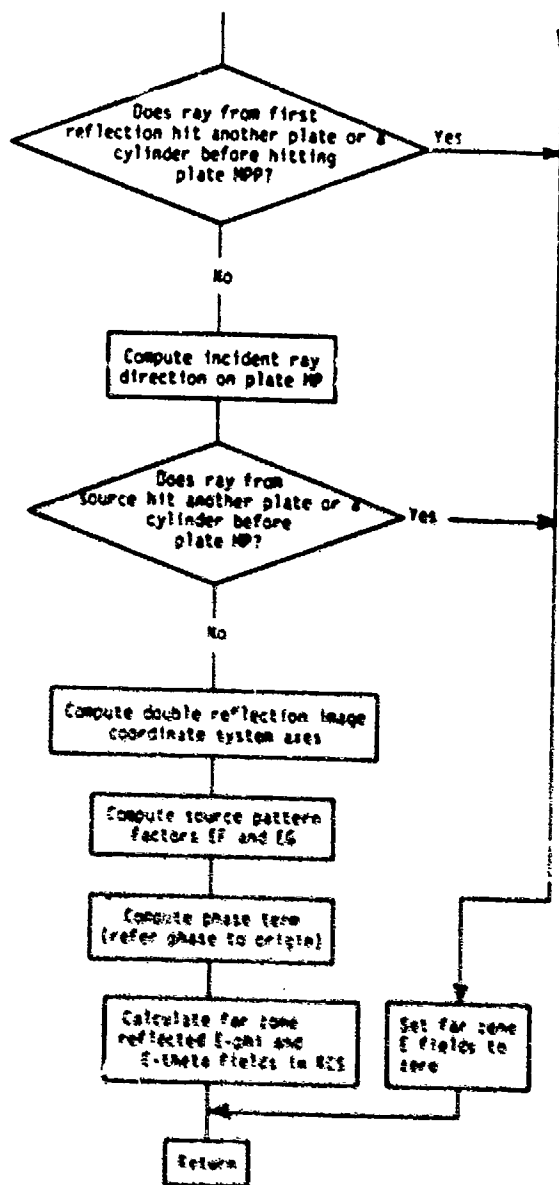
The doubly reflected fields are found using image theory. The double reflection source image is found so that the appropriate boundary conditions are met at the reflection points. The ray paths are checked to insure that they hit the appropriate plates and are not shadowed by other obstacles. The phase factor, $e^{jkD \cdot \hat{X} \hat{Y}}$, is then added to the pattern factor obtained from the SOURCE subroutine. The doubly reflected field is given in the form

$$E^{rr}(r, \theta, \phi) = W_m (ERT\hat{\theta} + ERP\hat{\phi}) \frac{e^{-jkR}}{R} .$$

The factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM





SYMBOL DICTIONARY

CPHI COSINE OF PHIR
 CPHJ COSINE OF PHJR
 CPHS COSINE OF PHSP
 CTHI COSINE OF THIR
 CTHJ COSINE OF THJR
 CTHS COSINE OF THSR
 D X,Y,Z COMPONENTS OF RAY PROPAGATION DIRECTION
 AFTER SECOND REFLECTION IN RCS
 DHIJ DISTANCE FROM DOUBLE REFLECTION IMAGE TO HIT POINT
 ON PLATE MPP
 DHIS DISTANCE BETWEEN REFLECTION POINTS
 DHIT DISTANCE FROM SOURCE TO REFLECTION POINT
 (FROM PLAIN)
 DI X,Y,Z COMPONENTS OF INCIDENT RAY PROPAGATION
 DIRECTION IN RCS
 DJ X,Y,Z COMPONENTS OF PROPAGATION DIRECTION
 OF RAY INCIDENT ON PLATE MPP
 EX COMPLEX PHASE FACTOR (CEXP(J*TPI*GAM))
 GAM PHASE DISTANCE TO ORIGIN (DOT PRODUCT OF DOUBLE
 REFLECTION IMAGE LOCATION AND REFLECTED RAY PROPAGATION
 DIRECTION)
 LHIT SET TRUE IF RAY INTERSECTS A PLATE OR CYLINDER
 (FROM PLAIN OR CYLIND)
 MP PLATE FROM WHICH FIRST REFLECTION OCCURS
 MPP PLATE FROM WHICH SECOND REFLECTION OCCURS
 PHIR PHI COMPONENT OF INCIDENT RAY PROPAGATION
 DIRECTION IN RCS
 PHJR PHI COMPONENT OF RAY DIRECTION BETWEEN REFLECTIONS IN RCS
 PHSR PHI COMPONENT OF RAY PROPAGATION DIRECTION
 AFTER REFLECTION IN RCS
 SPHI SINE OF PHIR
 SPHJ SINE OF PHJR
 SPHS SINE OF PHSR
 STHI SINE OF THIR
 STHJ SINE OF THJR
 THIR THETA COMPONENT OF INCIDENT RAY PROPAGATION
 DIRECTION IN RCS
 THJR THETA COMPONENT OF RAY DIRECTION BETWEEN REFLECTIONS IN RCS
 THSR THETA COMPONENT OF RAY PROPAGATION DIRECTION
 AFTER REFLECTIONS IN RCS
 VAX X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF THE
 SOURCE IMAGE COORDINATE SYSTEM AXES IN RCS COMPONENTS
 VAXP X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF THE
 SOURCE IMAGE COORDINATE SYSTEM AXES IN RCS
 FOR DOUBLE REFLECTION
 XI TRIPLY DIMENSIONED ARRAY OF IMAGE LOCATIONS
 XIJ X,Y,Z COMPONENTS OF DOUBLE REFLECTION IMAGE LOCATION
 XIS X,Y,Z COMPONENTS OF SINGLE REFLECTION SOURCE IMAGE LOCATION
 (SINGLE REFLECTION FROM PLATE MP)
 XS SOURCE LOCATION IN (X,Y,Z) RCS

CODE LISTING

```

1 C-----
2 SUBROUTINE RPLRPL(ERT,ERP,MP,MPP)
3 C!!!
4 C!!! DETERMINES THE REFL./REFL. FIELD WITH PHASE REFERRED TO
5 C!!! ORIGIN. RAY IS REFL. BY PLATE#MP THEN BY PLATE#MPP.
6 C!!!
7 COMPLEX EF,EG,EX,ERT,ERP,EIX,EIY,EIZ
8 DIMENSION XIS(3),XIJ(3),DI(3),DJ(3),VAX(3,3),VAXP(3,3)
9 LOGICAL LHIT
10 LOGICAL LDEBUG,LTEST
11 COMMON/TEST/LDEBUG,LTEST
12 COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
13 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
14 2,MEP(14),MPX
15 COMMON/SORINF/XS(3),VXS(3,3)
16 COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
17 COMMON/PI/PI,TPI,DPR,RPD
18 IF (LDEBUG) WRITE (6,101)
19 101 FORMAT (/, 'DEBUGGING RPLRPL SUBROUTINE')
20 C!!! SPECIFY IMAGE POSITION AFTER DOUBLE REFL.
21 DO 5 N=1,3
22 5 XIJ(N)=XI(MP,MPP,N)
23 C!!! DOES RAY FROM DOUBLE REFL. IMAGE HIT PLATE #MPP
24 CALL PLAIN(XIJ,D,DHIJ,-MPP,LHIT)
25 IF(.NOT.LHIT) GO TO 50
26 C!!! DOES DOUBLE REFL. RAY HIT ANOTHER PLATE
27 CALL PLAIN(XIJ,D,DHT,MPP,LHIT)
28 IF(LHIT) GO TO 50
29 C!!! DOES DOUBLE REFL. RAY HIT A CYLINDER
30 CALL CYLINT(XIJ,D,PHSR,DHT,LHIT,.TRUE.)
31 IF(LHIT) GO TO 50
32 C!!! COMPUTE THE RAY DIR BETWEEN PLATES MP AND MPP (DJ)
33 CALL REFBP(PHJR,THJR,PHSR,THSR,MPP)
34 IF (LDEBUG) WRITE (6,*) PHJR,THJR,PHSR,THSR,MPP
35 SPHJ=SIN(PHJR)
36 CPHJ=COS(PHJR)
37 STHJ=SIN(THJR)
38 CTHJ=COS(THJR)
39 DJ(1)=CPHJ*STHJ
40 DJ(2)=SPHJ*STHJ
41 DJ(3)=CTHJ
42 C!!! SPECIFY IMAGE LOCATION FOR SOURCE AFTER FIRST REFLECTION
43 DO 6 N=1,3
44 6 XIS(N)=XI(MP,MP,N)
45 C!!! DOES RAY FROM FIRST IMAGE HIT PLATE #MP
46 CALL PLAIN(XIS,DJ,DHIT,-MP,LHIT)
47 IF(.NOT.LHIT) GO TO 50
48 DHIS=DHIJ-DHIT
49 DHIS=DHIS-1.E-3
50 C!!! DOES RAY FROM FIRST IMAGE HIT ANOTHER PLATE BEFORE PLATE MPP?
51 CALL PLAIN(XIS,DJ,DHT,MP,LHIT)
52 IF(LHIT.AND.(DHT.LT.DHIS)) GO TO 50
53 C!!! DOES RAY HIT A CYLINDER
54 CALL CYLINT(XIS,DJ,PHJR,DHT,LHIT,.TRUE.)
55 IF(LHIT.AND.(DHT.LT.DHIS)) GO TO 50
56 C!!! KNOWING RAD. DIRECTION COMPUTE INCIDENT DIRECTION
57 C!!! ON PLATE #MP
58 CALL REFBP(PHIR,THIR,PHJR,THJR,MP)
59 IF (LDEBUG) WRITE (6,*) PHIR,THIR,PHJR,THJR,MP
60 SPHI=SIN(PHIR)
61 CPHI=COS(PHIR)
62 STHI=SIN(THIR)
63 CTHI=COS(THIR)
64 DI(1)=CPHI*STHI
65 DI(2)=SPHI*STHI
66 DI(3)=CTHI

```

```

67 C!!! DOES RAY FROM SOURCE HIT ANOTHER PLATE BEFORE PLATE MP?
68 CALL PLAIN(XS,DI,DHT,MP,LHIT)
69 IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 50
70 C!!! DOES RAY FROM SOURCE HIT A CYLINDER
71 CALL CYLINT(XS,DI,PHIR,DHT,LHIT,.FALSE.)
72 IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 50
73 C!!! COMPUTE DOUBLE REFL. SOURCE IMAGE COORD SYS AXES
74 DO 40 NJ=1,3
75 DO 40 NI=1,3
76 40 VAX(NI,NJ)=VXI(NI,NJ,MP)
77 CALL IMDIR(VAXP,VAX,MPP)
78 C!!! IF REFL/REFL FIELD EXISTS COMPUTE THE SOURCE PATTERN FACTORS
79 CALL SOURCE(EF,EG,EIX,EIY,EIZ,THSR,PHSR,VAXP)
80 IF (LDEUG) WRITE (6,*) EF,EG
81 C!!! COMPUTE PHASE REFERRED TO ORIGIN
82 GAM=XI(MP,MPP,1)*D(1)+XI(MP,MPP,2)*D(2)+XI(MP,MPP,3)*D(3)
83 EX=CEXP(CMPLX(0.,TPI*GAM))
84 C!!! CALCULATE FAR-ZONE E-PHI AND E-THETA FIELDS
85 ERT=EF*EX
86 ERP=EG*EX
87 GO TO 1
88 50 CONTINUE
89 ERT=(0.,0.)
90 ERP=(0.,0.)
91 1 IF (.NOT.LTEST) GO TO 2
92 WRITE (6,3)
93 3 FORMAT (/, ' TESTING RPLRPL SUBROUTINE' )
94 WRITE (6,*) ERT,ERP,MP,MPP
95 2 RETURN
96 END

```

RPLSCL

PURPOSE

To calculate the far-zone electric field of a source ray which is reflected by a given plate and then scattered by the cylinder.

PERTINENT GEOMETRY

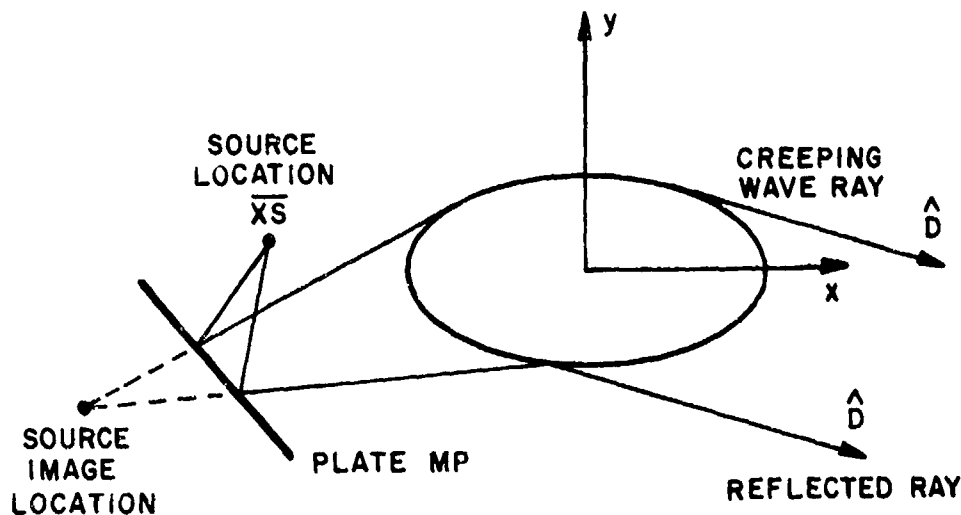


Figure 101--Illustration of ray reflected by a plate and then scattered by the cylinder.

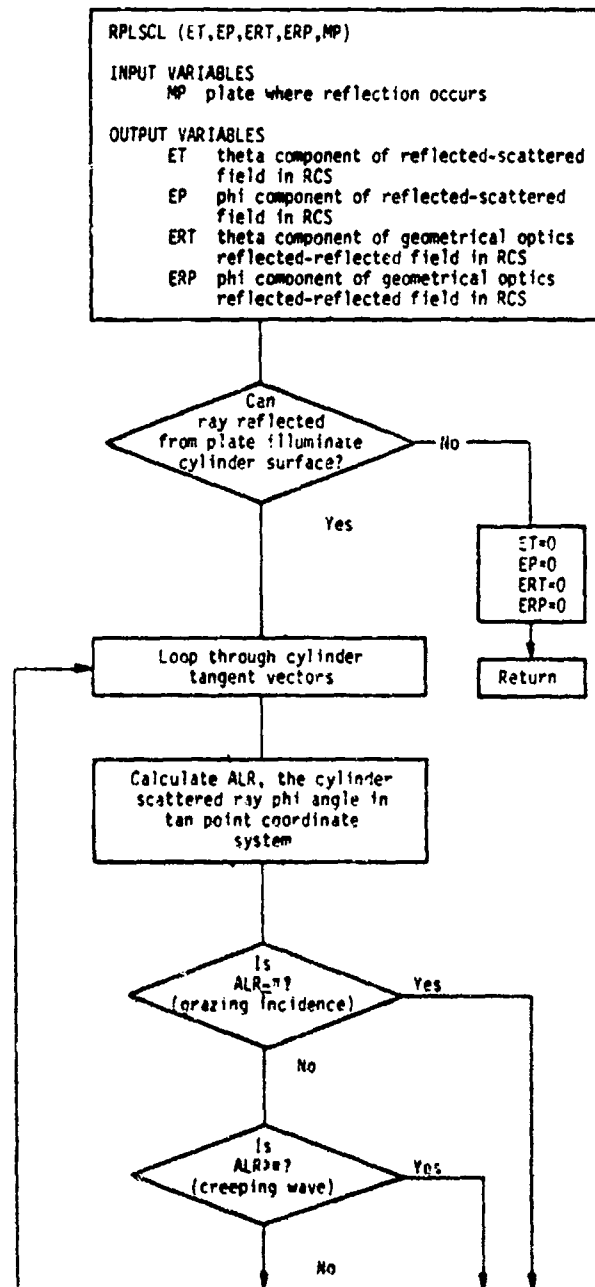
METHOD

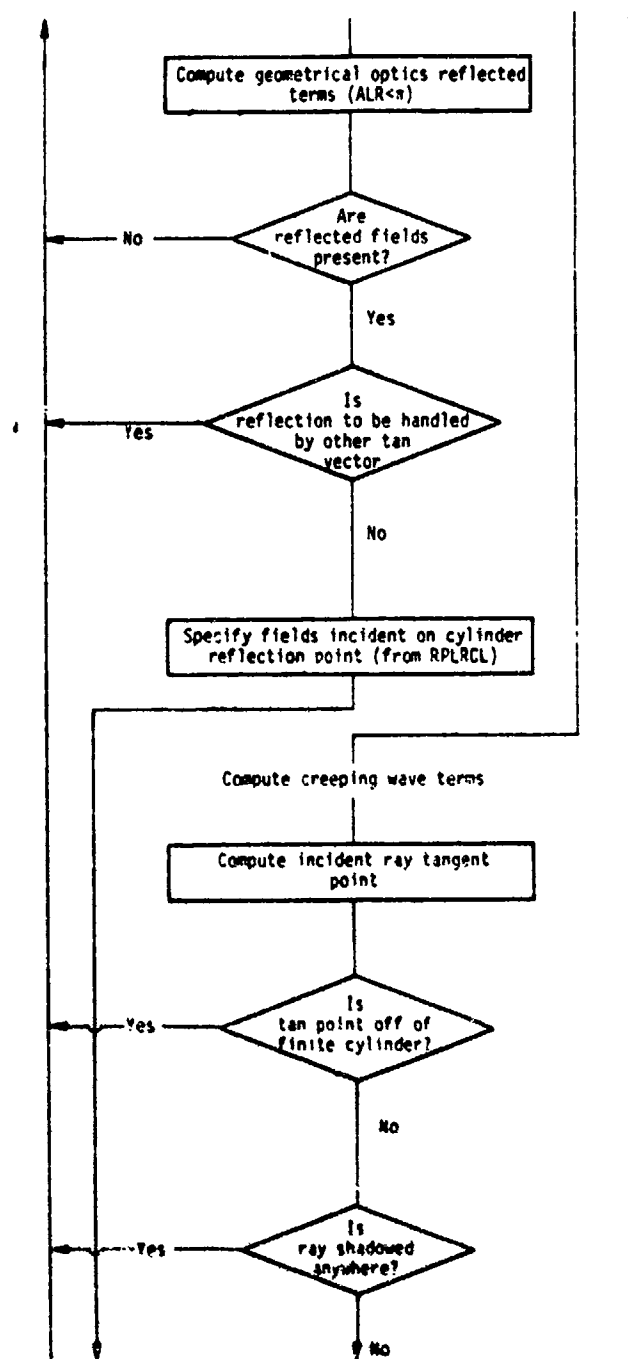
A uniform Geometrical Theory of Diffraction solution for the field reflected by a plate then reflected or diffracted by a cylinder is computed in this subroutine. The fields reflected or diffracted by the cylinder are determined in a similar manner as the fields calculated in subroutine SCTCYL. However, the incident field is found from the image source for the particular plate of interest, as illustrated in Figure 101. The image source fields are calculated in a similar manner to those obtained in subroutine REFPLA. The phase of the resultant reflected-scattered fields are referred to the reference coordinate system origin. The form of this field is then given by

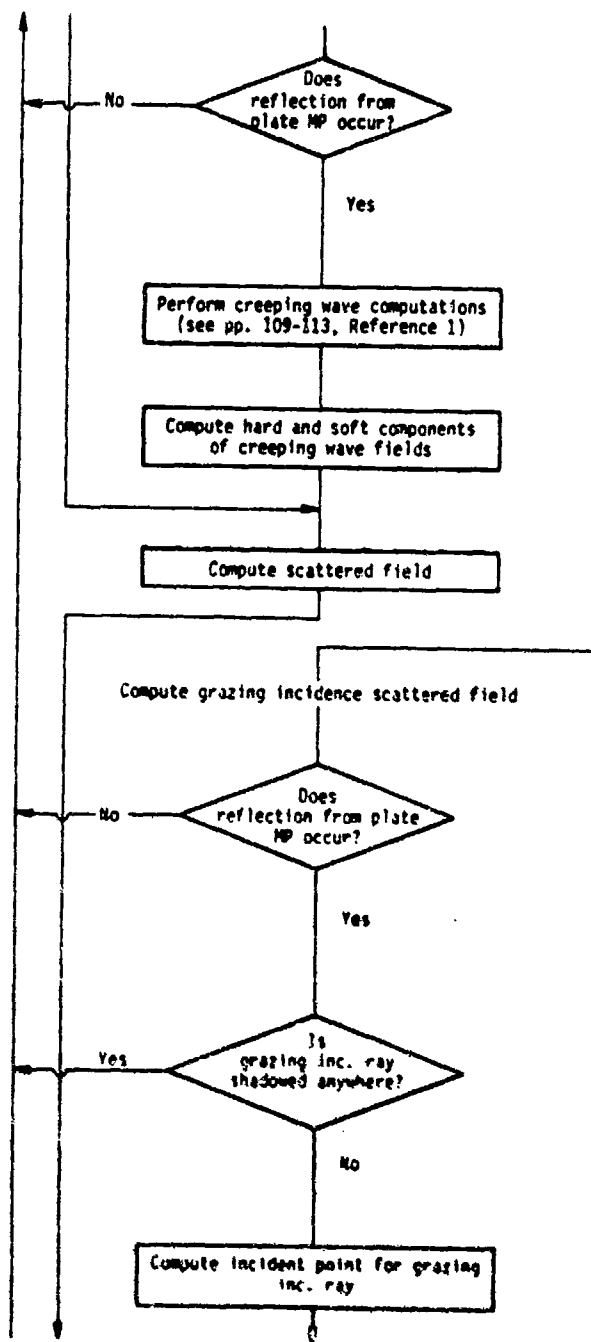
$$E^{r,s} = W_m (ET\hat{\theta} + EP\hat{\phi}) \frac{e^{-jkR}}{R} .$$

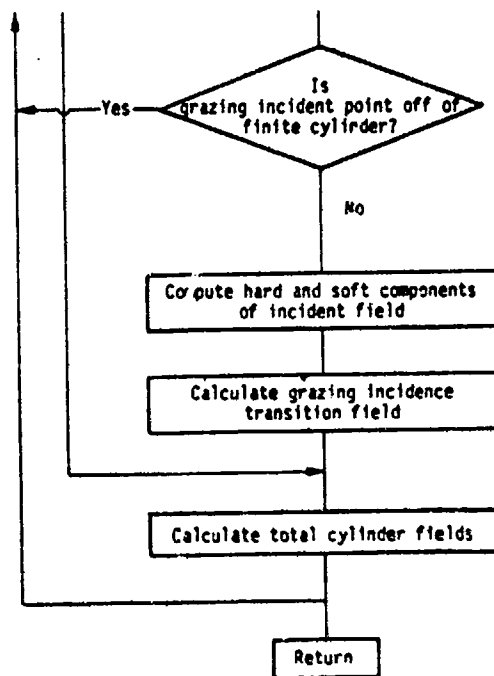
where the factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM









SYMBOL DICTIONARY

ALH	CYLINDER REFLECTED RAY PHI ANGLE IN TAN POINT COORDINATE SYSTEM (2-D)
ALS	PHI ANGLE DEFINING DIRECTION OF RAY FROM RCS ORIGIN TO SOURCE IMAGE IN TAN POINT COORD SYS
BX	X,Y,Z COMPONENTS OF POLARIZATION UNIT VECTOR OF SOFT COMPONENT OF FIELD INCIDENT ON CYL (PARALLEL TO CYL SURFACE AND NORMAL TO INC RAY PROP DIR)
BY	
BZ	
CFH	HARD TRANSITION FIELD COEFFICIENT
CFS	SOFT TRANSITION FIELD COEFFICIENT
DEPH	PHI COMPONENT OF TRANSITION FIELD IN RCS
DETH	THETA COMPONENT OF TRANSITION FIELD IN RCS
DHIT	DISTANCE FROM SOURCE IMAGE TO PLATE REFLECTION POINT (FROM PLAIN)
DHIV	DISTANCE FROM PLATE REFLECTION POINT TO CYLINDER
DHI	DISTANCE FROM SOURCE TO HIT POINT (FROM PLAIN)
DI	UNIT VECTOR OF RAY INCIDENT ON CYLINDER
DJ	X,Y,Z COMPONENTS OF UNIT VECTOR OF PROPAGATION DIRECTION OF SOURCE RAY INCIDENT ON PLATE
EF	PATTERN FACTOR FOR THETA COMPONENT OF INCIDENT FIELD IN RCS
EG	PATTERN FACTOR FOR PHI COMPONENT OF INCIDENT FIELD IN RCS
EHP	PHI COMPONENT OF HARD COMPONENT OF FIELD INCIDENT ON CYLINDER IN RCS
ENT	THETA COMPONENT OF HARD COMPONENT OF FIELD INCIDENT ON CYLINDER IN RCS
EP	PHI COMPONENT OF CYLINDER SCATTERED E FIELD WITH PHASE REFERRED TO RCS ORIGIN
ER	DOT PRODUCT OF UNIT VECTOR TANGENT TO CYLINDER AND THE PROPAGATION DIR. UNIT VECTOR
ESP	PHI COMPONENT OF SOFT COMPONENT OF FIELD INCIDENT ON CYLINDER IN RCS
EST	THETA COMPONENT OF SOFT COMPONENT OF FIELD INCIDENT ON CYLINDER IN RCS
ET	THETA COMPONENT OF CYLINDER SCATTERED E FIELD WITH PHASE REFERRED TO RCS ORIGIN
EX	PATTERN FACTOR FOR X,Y,Z COMPONENTS OF INCIDENT FIELD IN RCS
EY	
EZ	
I	VARIABLE USED TO STEP THROUGH TANGENT POINTS
LHIT	SET TRUE IF RAY HITS A PLATE (FROM PLAIN)
LTRFI	(RETURNED FROM RPLRCL) SET TRUE IF G.O. CYLINDER REFLECTED FIELD DOES NOT EXIST
PHIR	PHI COMPONENT OF PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER
PHJR	PHI COMPONENT OF PROPAGATION DIRECTION OF SOURCE RAY INCIDENT ON PLATE
S	LENGTH OF VECTOR FROM SOURCE IMAGE TO TAN POINT (2 OR 3-D)
THIR	THETA COMPONENT OF PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER
THJR	THETA COMPONENT OF PROPAGATION DIRECTION OF SOURCE RAY INCIDENT ON PLATE
VL	ELL ANGLE DEFINING POINT WHERE CREEPING WAVE LEAVES CYLINDER
VI	ELL. ANGLE USED TO DEFINE TANGENT POINTS (2-D)
VL	ELL ANGLE DEFINING LOWER RANGE OF CREEPING WAVE TRAVEL ON CYLINDER (2-D)
VC	ELL ANGLE DEFINING UPPER RANGE OF CREEPING WAVE TRAVEL ON CYLINDER (2-D)
XD	X,Y,Z COMPONENTS OF DIRECTION OF RAY FROM SOURCE TO CYLINDER TANGENT POINT (INCIDENT RAY FOR CREEPING AND GRAZING INC. CASES)
YD	
ZD	

XII	}	X,Y,Z COMPONENTS OF POINT WHERE INCIDENT CREEPING	
YII			WAVE (OR GRAZING WAVE) MEETS CYLINDER
ZII			
XIS		X,Y,Z COMPONENTS OF IMAGE SOURCE LOCATION (FOR	
		REFLECTION FROM PLATE MP)	
APP		X,Y,Z COMPONENTS OF POINT WHERE RAY LEAVES CYLINDER	
AMF		X,Y,Z COMPONENTS OF POINT WHERE CREEPING WAVE	
		LEAVES CYLINDER	

CODE LISTING

```

1 C-----
2 SUBROUTINE RPLSCL(ET,EP,ERT,ERP,MP)
3 C!!!
4 C!!! COMPUTES THE FIELD REFLECTED FROM PLATE #MP THEN
5 C!!! SCATTERED FROM THE ELLIPTIC CYLINDER
6 C!!!
7 COMPLEX CJ,CPI4,CF,CFH,CFS,F1,PFUN,OFUN
8 COMPLEX EIX,EIY,EIZ,EIPH,EITH,ET,EP,ERT,ERP
9 COMPLEX REF,ESTH,ESPH,ENTH,ENPH,DETH,DEPH,EF,EG
10 COMPLEX EST,ESP,EHT,EHP
11 DIMENSION VI(2),ER(2),UH(2),US(2),DI(3),XRF(3)
12 DIMENSION XIS(3),DJ(3),VAX(3,3)
13 LOGICAL LHIT,LTRFI,LDEBUG,LTEST,LRFI,LRFIT
14 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
15 COMMON/SORINF/XS(3),VXS(3,3)
16 COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
17 COMMON/PIS/PI,TPI,DPR,RPD
18 COMMON/GTD/AS,IS,SAS,SASP,CAS
19 COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
20 COMMON/COMP/CJ,CPI4
21 COMMON/BNDCI/DTI(14),VTI(14,2),BTI(14,4)
22 COMMON/FUDGI/REF,ESTH,ESPH,ENTH,ENPH,XE(3),RG,RIOI,OL,LTRFI
23 COMMON/TEST/LDEBUG,LTEST
24 COMMON/CLRFI/LRFI(14)
25 EXTERNAL FCT
26 IF(LDEBUG) WRITE(6,900)
27 900 FORMAT(/,' DEBUGGING RPLSCL SUBROUTINE')
28 ET=(0.,0.)
29 EP=(0.,0.)
30 EHT=(0.,0.)
31 ERP=(0.,0.)
32 C!!! CAN PLATE REFLECTED RAY ILLUMINATE CURVED SURFACE?
33 IF(DTI(MP).LT.-1.5) GO TO 909
34 ER(1)=BTI(MP,1)*CPS+BTI(MP,2)*SPS
35 ER(2)=BTI(MP,3)*CPS+BTI(MP,4)*SPS
36 C!!! LOOP THRU TANGENT VECTORS
37 J=1
38 LRFIT=.FALSE.
39 VTI(1)=VTI(MP,1)
40 VTI(2)=VTI(MP,2)
41 3 CLTINUE
42 CALL NANOBI(UH,US,VTI(1))
43 SINA=UH(1)*CPS+UH(2)*SPS
44 C!!! CALCULATE ALR, THE REFLECTED RAY PHI ANGLE IN
45 C!!! TANGENT POINT COORD. SYS.
46 ALR=ATAN2(SINA,-ER(1))
47 IF(ALR.LT.0.) ALR=ALR+PI
48 C!!! IF GRAZING INCIDENCE IS PRESENT, SKIP TO
49 C!!! APPROPRIATE SECTION
50 IF(ABS(PI-ALR).LT.2.0005) GO TO 5
51 C!!! IF ALR.GT.PI COMPUTE CREEPING WAVE TERMS
52 IF(ALR.GT.PI) GO TO 10
53 C!!! COMPUTE REFLECTED FIELD TERMS IF ALR .LE. PI
54 CALL WPLRCL(ENT,ERP,MP)
55 C!!! ARE REFLECTED FIELDS PRESENT?
56 IF(LTRFI) GO TO 1
57 SNAS=UH(1)*XI(MP,MP,1)+UH(2)*XI(MP,MP,2)
58 IC=2*1-1
59 CSAS=BTI(MP,IC)*VTI(MP,MP,1)+BTI(MP,IC+1)*VTI(MP,MP,2)
60 ALS=ATAN2(SNAS,-CSAS)
61 ALRS=ALS-ALS
62 C!!! IS REFLECTION TO BE HANDLED BY OTHER TANGENT VECTOR?
63 IF(ABS(ALRS).LT.0.0005.AND.1.EQ.2) GO TO 1
64 IF(ALRS.LE.-0.0005) GO TO 1
65 GR=1*(1+RG)/(1+RG)
66 GR5=GR*OL/(OL-2*RG)

```

```

67      SKWIG=-ABS(2.*TP1+RHS/GM/GM)
68      CF=-SORT(-2./PI/SKWIG)*CP14*REF
69      CF=CF*CEXP(-CJ*SKWIG*SKWIG*SKWIG/12.)
70      TTRM=SKWIG/GM
71      XX=PI*(DL+RHS)*TTRM*TTRM
72 C!!!  SPECIFY HARD AND SOFT COMPONENTS OF FIELD INC. ON CYLINDER
73 C!!!  FROM RPLRCL
74      EST=ESTH
75      ESP=ESPH
76      ENT=ENTH
77      EHP=EHPH
78      GO TO 30
79 10    CONTINUE
80      IF(LRFIT) LRFI(MP)=.FALSE.
81      LRFIT=.TRUE.
82 C!!!  COMPUTE CREEPING WAVE TERMS IF ALR .GT. PI
83 C!!!  COMPUTE INCIDENT RAY TANGENT POINT
84      XII=A*COS(VI(1))
85      YII=B*SIN(VI(1))
86      XD=XII-XI(MP,MP,1)
87      YD=YII-XI(MP,MP,2)
88      S=SORT(XD*XD+YD*YD)
89      ZII=S*CTHS/STHS*XI(MP,MP,3)
90 C!!!  IS TAN POINT ON (FINITE) CYLINDER?
91      IF(ZII.GT.ZC(1)+XII*CTC(1).OR.
92      2ZII.LT.ZC(2)+XII*CTC(2)) GO TO 1
93      ZD=ZII-XI(MP,MP,3)
94      PHIR=BTAN2(YD,XD)
95      THIR=BTAN2(S,ZD)
96      S=SORT(S*S+ZD*ZD)
97      DI(1)=XD/S
98      DI(2)=YD/S
99      DI(3)=ZD/S
100     DO 15 N=1,3
101 15    XIS(N)=XI(MP,MP,N)
102 C!!!  DOES REFLECTION OFF OF PLATE MP OCCUR?
103     CALL PLAIN(XIS,DI,DHIT,MP,LHIT)
104     IF(.NOT.LHIT) GO TO 1
105     DHIV=S-DHIT
106 C!!!  IS RAY SHADOWED BETWEEN REFLECTION AND DIFFRACTION?
107     CALL PLAIN(XIS,DI,DHT,MP,LHIT)
108     IF(LHIT.AND.(DHT.LT.DHIV)) GO TO 1
109 C!!!  CALCULATE PROPAGATION DIRECTION OF RAY INCIDENT
110 C!!!  ON PLATE MP
111     CALL REFR(PHJR,THJR,PHIR,THIR,MP)
112     SPHJ=SIN(PHJR)
113     CPHJ=COS(PHJR)
114     STHJ=SIN(THJR)
115     CTHJ=COS(THJR)
116     DJ(1)=CPHJ*STHJ
117     DJ(2)=SPHJ*STHJ
118     DJ(3)=CTHJ
119 C!!!  IS SOURCE RAY SHADOWED BEFORE HITTING PLATE MP?
120     CALL PLAIN(XS,DJ,DHT,MP,LHIT)
121     IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 1
122     CALL CYLINT(XS,DJ,PHJR,DHT,LHIT,.FALSE.)
123     IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 1
124     DO 20 NJ=1,3
125 C!!!  SPECIFY SOURCE IMAGE AXES AND CALCULATE
126 C!!!  SOURCE PATTERN FACTOR
127     DO 20 NI=1,3
128 20    VAXI(NI,NJ)=VXI(NI,NJ,MP)
129     CALL SOURCE(EP,EO,EI1,EI2,THIR,PHIR,VAXI
130 C!!!  PERFORM CREEPING WAVE COMPUTATIONS
131     IF(ILDERUG) WRITE(6,*) EP,EO
132     IF(1.EQ.1) VD=BTAN2(-D*CPJ,A*SPJ)

```

```

133      IF(I.EQ.2) VD=BTAN2(B=CPS,-A=SPS)
134      VDP=VD-VI(1)
135      IF(VDP.GT.PI) VDP=VDP-PI
136      IF(VDP.LT.-PI) VDP=VDP+PI
137      IF(I.EQ.2) GO TO 20
138      IF(VDP.LT.0.) GO TO 1
139      VL=VI(1)
140      VU=VDP+VI(1)
141      GO TO 25
142 20    CONTINUE
143      IF(VDP.GT.0.) GO TO 1
144      VL=VDP+VI(1)
145      VU=VI(1)
146 25    CONTINUE
147      CALL FKARG(SKNIG,AS,VL,VU)
148      XRF(1)=A*COS(VD)
149      XRF(2)=B*SIN(VD)
150      ID=3
151      CALL DOG32(VL,VU,FCT,SS)
152      SS=SS/SAS
153      XRF(3)=2*I+SS*CTHS
154 C!!!  DOES RAY HIT PLATE AFTER LEAVING CYLINDER?
155      CALL PLANT(XRF,D,DHIT,U,LHIT)
156      IF(LHIT) GO TO 1
157      CALL RADCV(RGI,RT,VI(1))
158      CALL RADCV(RGF,RT,VD)
159      GMM=(PI*PI*RG)*RGF**(.1/.4)
160      CF=GMM*CP14*CEXP(-CJ*TP1*(S+SS))/PI/SORT(2.*S)
161      CF=CF*CEXP(CJ*TP1*(XRF(1)*D(1)+XRF(2)*D(2)+XRF(3)*D(3)))
162      TTRM=SKNIG/GMM
163      XX=PI*S*TTRM*TTRM
164      BX=-UN(2)*D(3)
165      BY=UN(1)*D(3)
166      BZ=UN(2)*D(1)-UN(1)*D(2)
167      ESP=(0.,0.)
168      ENT=(0.,0.)
169 C!!!  COMPUTE HARD AND SOFT CREEPING WAVE COMPONENTS
170      EHP=EIX*UN(1)+EY*UN(2)
171      EST=EIX*BX+EY*BY+EZ*BZ
172      IF(I.EQ.1) EHP=-EHP
173      IF(I.EQ.2) EST=-EST
174 30    CONTINUE
175 C!!!  COMPUTE THE SCATTERED FIELD
176      XIS=SORT(TPI*XI)
177      XI=SORT(2.*XI/PI)
178      CALL FRNLS(CCC,SSS,XI)
179      FI=CPPL1(0.5-CCC,SSS-0.5)
180      FI=XIS*FI*CEXP(CJ*(1.5*PI*XI))
181      FI=-FI/SKNIG/SORT(2.)
182      SOTF=SORT(2.*PI)
183      CFH=CF*(PI*SOTF*GMM*UN(SKNIG))
184      CFS=CF*(PI*SOTF*PI*UN(SKNIG))
185      DEPH=CFH*EHP+CFS*ESP
186      DETH=CFH*ENT+CFS*EST
187      GO TO 0
188 5     CONTINUE
189 C!!!  COMPUTE GRAZING INC. SCATTERED FIELD
190      DO 35 N=1,3
191 25    A1=BNIT(NP,NP,NI)
192 C!!!  DOES REFLECTION FROM PLATE NP OCCUR IN GRAZING INCIDENCE
193      DIRECTION?
194      CALL PLANT(NIS,D,DHIT,-NP,LHIT)
195      IF(.NOT.LHIT) GO TO 1
196 C!!!  IS RAY TRANSMITTED BETWEEN REFLECTIONS?
197      CALL PLANT(NIS,D,DPT,NP,LHIT)
198      IF(LHIT) GO TO 1

```

```

199      CALL REFBP(PHJH,THJR,PHSR,THSR,MP)
200      SPHJ=SIN(PHJR)
201      CPHJ=COS(PHJR)
202      STHJ=SIN(THJR)
203      CTHJ=COS(THJR)
204      DJ(1)=CPHJ*STHJ
205      DJ(2)=SPHJ*STHJ
206      DJ(3)=CTHJ
207 C!!! IS INCIDENT (SOURCE) RAY SHADOWED BY PLATE OR CYLINDER?
208      CALL PLAIN(XS,DJ,DHT,MP,LHIT)
209      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 1
210      CALL CYLINT(XS,DJ,PHJH,DHT,LHIT,.FALSE.)
211      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 1
212 C!!! CALCULATE GRAZING INCIDENCE POINT
213      SGN=-SIGN(1.,SIN(ALR))
214      XII=A*COS(VI(1))
215      YII=B*SIN(VI(1))
216      XD=XII-XI(MP,MP,1)
217      YD=YII-XI(MP,MP,2)
218      S=SQRT(XD*XD+YD*YD)
219      ZII=S*CTHS/STHS*XI(MP,MP,3)
220 C!!! IS GRAZING INC. POINT OFF OF FINITE CYLINDER?
221      IF(ZII.GT.ZC(1)+XII*CTC(1).OR.
222      2ZII.LT.ZC(2)+XII*CTC(2)) GO TO 1
223      ZD=ZII-XI(MP,MP,3)
224      S=SQRT(S*S+ZD*ZD)
225      CALL RADCV(RGI,RT,VI(1))
226      GM=(PI*RCI)**(1./3.)
227      DO 36 NJ=1,3
228      DO 36 NI=1,3
229 36      VAX(NI,NJ)=VXI(NI,NJ,MP)
230      CALL SOURCE(EP,EG,EIX,EIY,EIZ,THSR,PHSR,VAX)
231      CF=CEXP(CJ)*PI*(XI(MP,MP,1)*D(1)+XI(MP,MP,2)*D(2)+
232      2XI(MP,MP,3)*D(3))
233      BX=-UN(2)*D(1)
234      BY=UN(1)*D(3)
235      BZ=UN(2)*D(1)-UN(1)*D(2)
236 C!!! CALCULATE HARD AND SOFT COMPONENTS OF INCIDENT FIELD
237      EHP=EIX*UN(1)+EIY*UN(2)
238      EST=EIX*BX+EIY*BY+EIZ*BZ
239      IF(1.EQ.1) EHP=-EHP
240      IF(1.EQ.2) EST=-EST
241      CFH=GM*CFI+CFUN(3.)/SQRT(PI*S)
242      CFS=GM*CFI+CFUN(0.)/SQRT(PI*S)
243 C!!! CALCULATE GRAZING INCIDENCE TRANSITION FIELD
244      DETH=(0.5*EP*SGN-CFS*EST)*CF
245      DEPH=(0.5*EG*SGN-CFH*EHP)*CF
246 6      CONTINUE
247 C!!! CALCULATE TOTAL CYLINDER FIELDS
248      EP=EP+DEPH
249      ET=ET+DETH
250      IF(1.DEBUG) WRITE(6,*) 1,SGN,G,XI,FI,CF
251      IF(1.DEBUG) WRITE(6,*) CFH,CFS
252      IF(1.DEBUG) WRITE(6,*) EHP,EST
253      IF(1.DEBUG) WRITE(6,*) EHP,ESP
254      IF(1.DEBUG) WRITE(6,*) DETH,DEPH
255 1      I=I+1
256      IF(1.LE.2) GO TO 3
257 409      CONTINUE
258      IF(.NOT.LTEST) RETURN
259      WRITE(6,910)
260 910      FORMAT(1,' TESTING MPLSCL SUBROUTINE:')
261      WRITE(6,*) ET,EP,MP
262      WRITE(6,*) EST,ESP
263      RETURN
264      END

```

SCLRPL

PURPOSE

To compute the far-zone electric field of a source ray which is scattered by the cylinder and then reflected by a given plate.

PERTINENT GEOMETRY

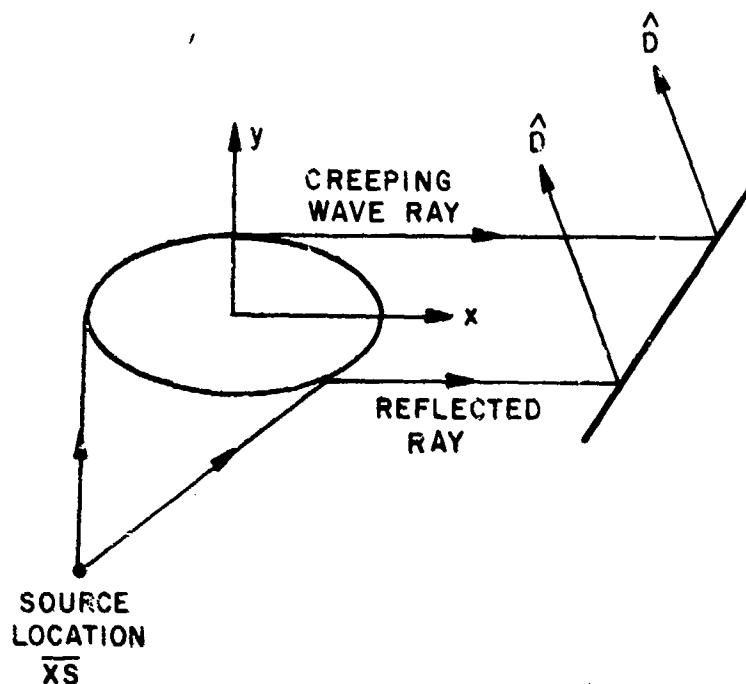


Figure 102--Illustration of ray scattered by the cylinder and reflected by a plate

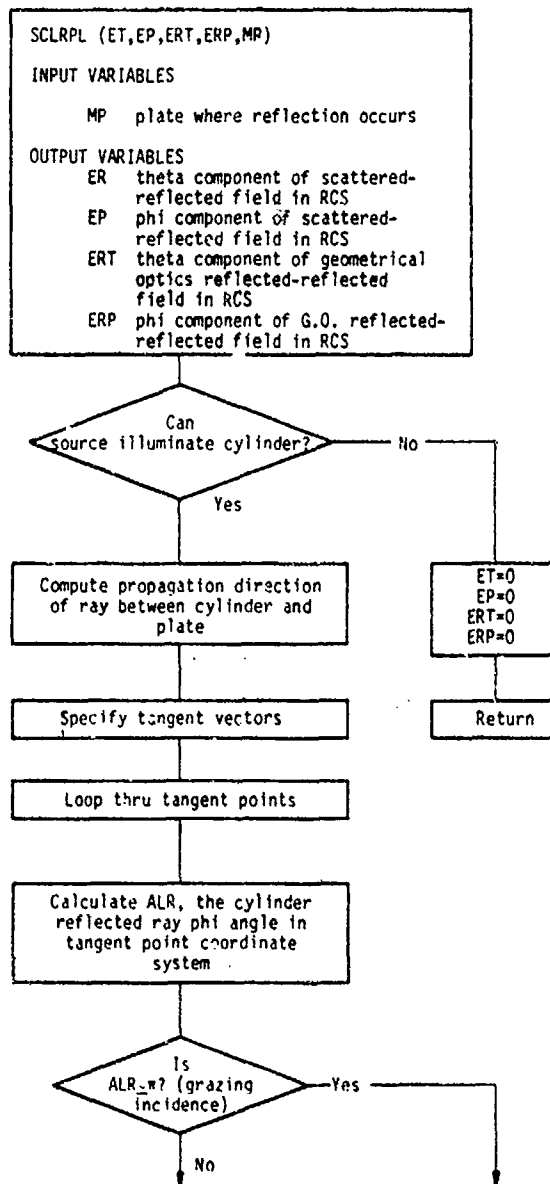
METHOD

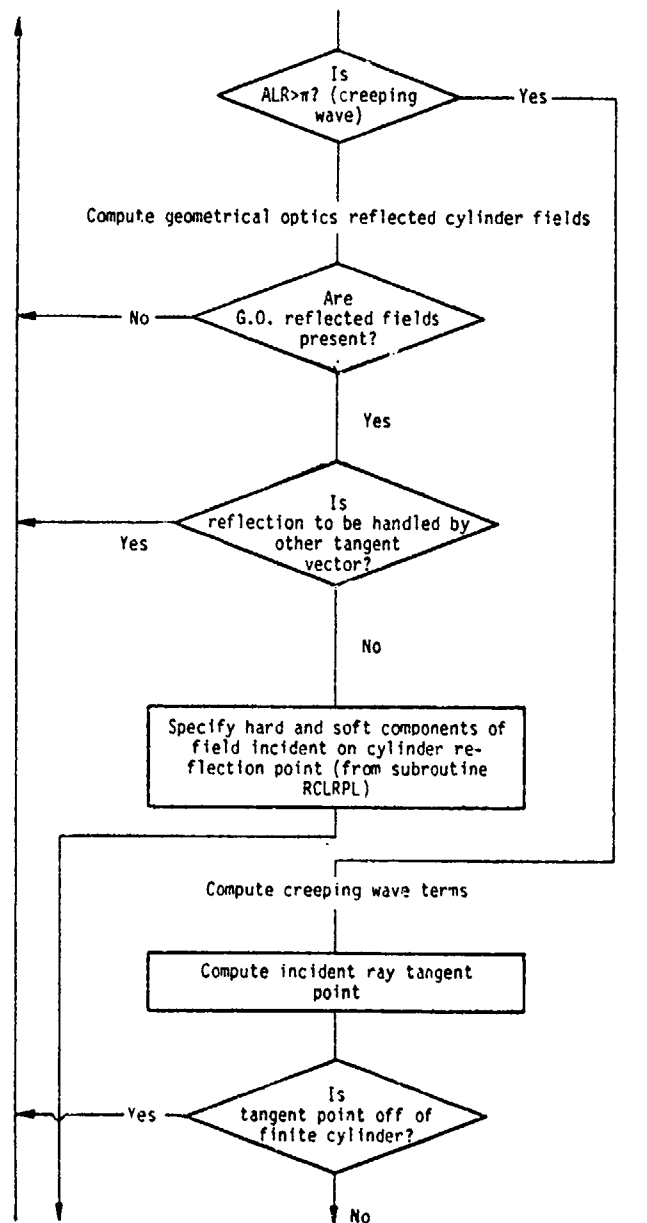
A uniform Geometrical Theory of Diffraction solution for the field reflected or diffracted by a cylinder then reflected by a plate is computed in this subroutine. The fields reflected or diffracted by the cylinder in the direction of the plate are determined in a similar manner as the fields calculated in subroutine SCTCYL. The direction of the ray incident on the plate is determined by imaging the observation direction into the plate, as illustrated in Figure 102. The plate reflected fields are found by satisfying the boundary conditions for the fields at the surface of the plate. The phase of the resultant scattered-reflected fields are referred to the reference coordinate origin. The form of this field is then given by

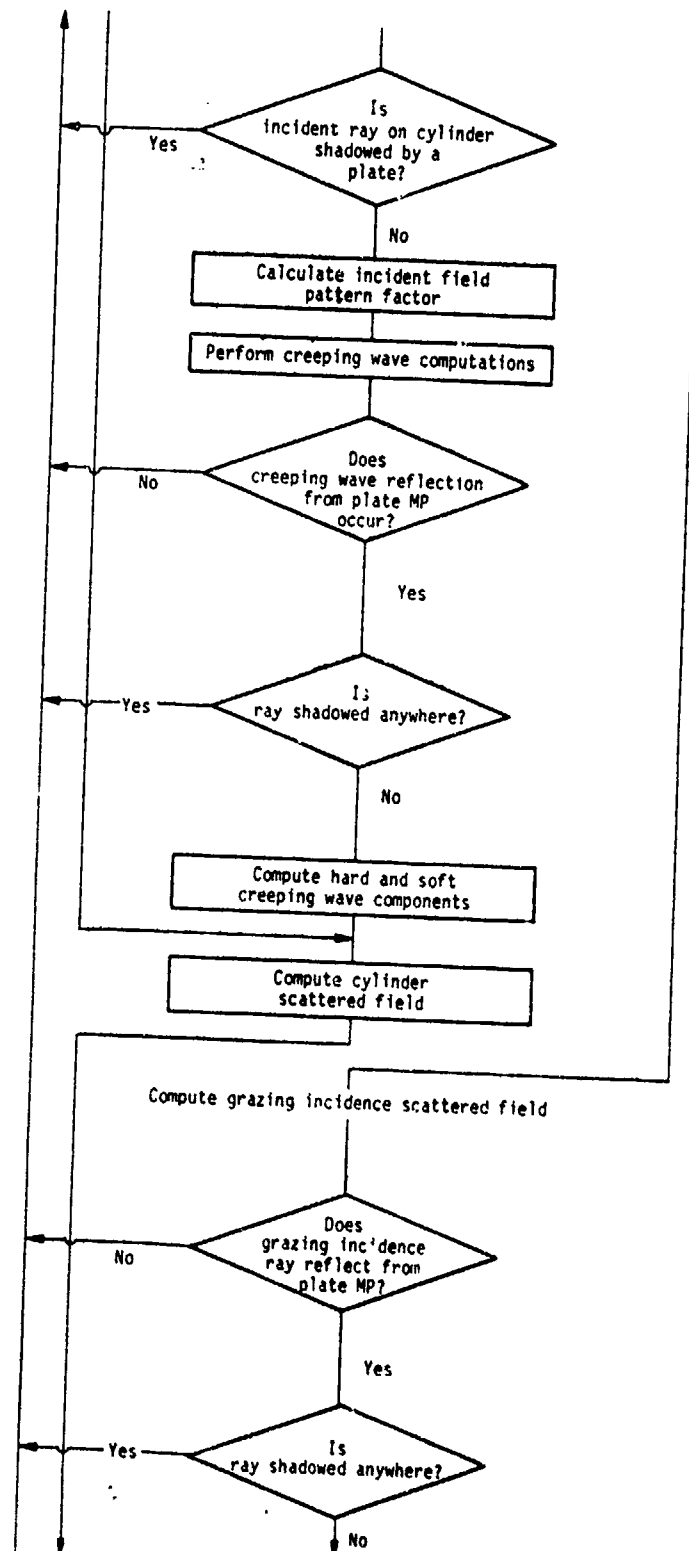
$$\vec{E}^{s,r} = W_m (ET\hat{\theta} + EP\hat{\phi}) \frac{e^{-jkR}}{R},$$

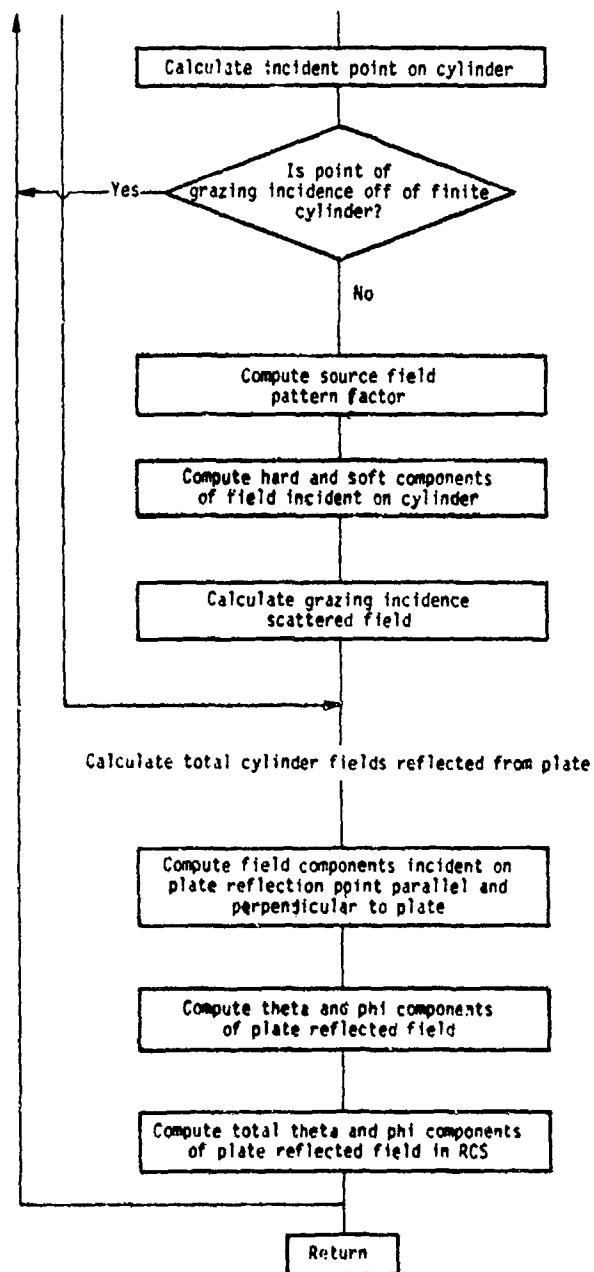
where the factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM









SYMBOL DICTIONARY

AI	}	FIELD COMPONENTS OF RAY INCIDENT ON PLATE
A2		NORMAL AND TANGENT TO PLATE
A3		DETERMINANT OF POLARIZATION TRANSFORMATION
ALH		PHI ANGLE DEFINING PROPAGATION DIRECTION IN TAN POINT COORDINATE SYSTEM (2-D)
ALHS		DIFFERENCE BETWEEN ALS AND ALR
ALS		PHI ANGLE DEFINING DIRECTION OF RAY FROM RCS ORIGIN TO SOURCE IN TAN POINT COORD SYS
BX	}	X,Y,Z COMPONENTS OF POLARIZATION UNIT VECTOR
BY		OF SOFT COMPONENT OF FIELD INCIDENT ON CYL (PARALLEL TO CYL SURFACE AND NORMAL TO INC FIELD PROP DIR)
BZ		
C11	}	COEFFICIENTS USED TO CONVERT POLARIZATION FROM THEIA AND PHI COMPONENTS IN RCS TO COMPONENTS NORMAL AND TANGENT TO PLATE (AND VICE-VERSA)
C12		
C21		
C22		
CFH		HARD TRANSITION FIELD COEFFICIENT
CFS		SOFT TRANSITION FIELD COEFFICIENT
DEPH		PHI COMPONENT OF TRANSITION FIELD IN RCS
DETH		THEIA COMPONENT OF TRANSITION FIELD IN RCS
DI		X,Y, AND Z COMPONENTS OF INCIDENT RAY DIRECTION ON CYL IN RCS
DJ		X,Y,Z COMPONENTS OF PROPAGATION DIRECTION OF RAY BETWEEN CYLINDER AND PLATE IN RCS
EF		THEIA COMPONENT OF SOURCE FIELD PATTERN FACTOR IN RCS
EG		PHI COMPONENT OF SOURCE FIELD PATTERN FACTOR IN RCS
EHP		PHI COMPONENT OF HARD COMPONENT OF GEOMETRICAL OPTICS FIELD INCIDENT ON CYLINDER IN RCS
ENT		THEIA COMPONENT OF HARD COMPONENT OF GEOMETRICAL OPTICS FIELD INCIDENT ON CYLINDER IN RCS
ER		DOT PRODUCT OF CYLINDER TANGENT UNIT VECTOR AND REFLECTED RAY PROPAGATION DIRECTION (2-D)
ERP		PHI COMPONENT OF G.O. REFL-REFL FIELD IN RCS
ERI		THEIA COMPONENT OF G.O. REFL-REFL FIELD IN RCS
ESP		PHI COMPONENT OF SOFT COMPONENT OF GEOMETRICAL OPTICS FIELD INCIDENT ON CYLINDER IN RCS
ESI		THEIA COMPONENT OF SOFT COMPONENT OF GEOMETRICAL OPTICS FIELD INCIDENT ON CYLINDER IN RCS
PHIR		PHI COMPONENT OF INCIDENT RAY DIRECTION ON CYL
PHJR		PHI COMPONENT OF RAY PROPAGATION DIRECTION BETWEEN CYLINDER AND PLATE
SKWIG		PARAMETER USED IN TRANSITION FUNCTION
THIR		THEIA COMPONENT OF INCIDENT RAY DIRECTION ON CYLINDER
THJR		THEIA COMPONENT OF RAY PROPAGATION DIRECTION BETWEEN CYLINDER AND PLATE
TIKM		PARAMETER USED IN TRANSITION FUNCTION
TX1	}	X,Y COMPONENTS OF RAY FROM SOURCE
TY1		TANGENT TO TAN POINT 1 (2-D)
TX2	}	X,Y COMPONENTS OF RAY FROM SOURCE
TY2		TANGENT TO TAN POINT 2 (2-D)
UB		X,Y COMPONENTS OF UNIT VECTOR TANGENT TO CYL AT TAN POINT
UN		X,Y COMPONENTS OF UNIT VECTOR NORMAL TO CYL AT TAN POINT
VI		ELL. ANGLE USED TO DEFINE TANGENT POINTS (2-D)
VL		ELL ANGLE DEFINING LOWER LIMIT OF CREEPING WAVE TRAVEL ON CYLINDER
V1		X,Y,Z COMPONENTS OF POLARIZATION UNIT VECTOR PERPENDICULAR TO PLANE OF INCIDENCE FOR RAY INCIDENT ON PLATE
VU		ELL ANGLE DEFINING UPPER LIMIT OF CREEPING WAVE TRAVEL ON CYLINDER
X0	}	X,Y,Z COMPONENTS OF DIRECTION OF RAY FROM SOURCE TO CYLINDER TANGENT POINT (INCIDENT RAY FOR CREEPING AND GRAZING INC. CASES)
Y0		
Z0	}	
X1		
Y1		X,Y,Z COMPONENTS OF POINT WHERE INCIDENT CREEPING WAVE (OR GRAZING WAVE) MEETS CYLINDER
Z1		
XRP		X,Y,Z COMPONENTS OF POINT WHERE CREEPING WAVE LEAVES CYLINDER
XNS		X,Y,Z COMPONENTS OF REFLECTION POINT LOCATION ON PLATE AND ALSO POINT WHERE CREEPING WAVE LEAVES CYLINDER ALSO IMAGE OF XRP IN PLATE PD

CODE LISTING

```

1 C-----
2 SUBROUTINE SCLRPL(ET,EP,ENT,ERP,MP)
3 C!!!
4 C!!! COMPUTES THE FIELD SCATTERED FROM THE CYLINDER THEN REFLECTED
5 C!!! FROM PLATE #MP
6 C!!!
7 COMPLEX CJ,CPI4,CF,CFH,CFS,FI,PFH,CFH
8 COMPLEX EIX,EIY,EIZ,EIPH,EITH,ET,EP,ENT,ERP
9 COMPLEX REF,ESTH,ESPH,EHTH,EHPH,DETH,DEPH,EF,EG
10 COMPLEX EST,ESP,ENT,EHP,A1,A2
11 DIMENSION VI(2),ER(2),UN(2),UB(2),DI(3),XRF(3),XRS(3),VT(3),DJ(3)
12 LOGICAL LHIT,LTRFJ,LDERUG,LTEST,LRF5,LRFST
13 COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
14 2,MEP(14),4PX
15 COMMON/GEOMEL/A,R,ZC(2),SNC(2),CNC(2),CTC(2)
16 COMMON/SORINF/XS(3),VXS(3,3)
17 COMMON/FIS/PI,TPI,DPH,RPD
18 COMMON/GTD/AS,IO,SAS,SASP,CAS
19 COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,SPIS,CTHS
20 COMMON/LHPHUV/DT(3),DP(2)
21 COMMON/COMP/CJ,CPI4
22 COMMON/LNDSCL/DTS,VTS(2),HTS(4)
23 COMMON/FUDGJ/REF,ESTH,ESPH,EHTH,EHPH,XR(3),RC,MMOI,DL,LTRFJ
24 COMMON/TEST/LDEBUG,LTEST
25 COMMON/CLRFS/LRFS(14)
26 EXTERNAL FCT
27 ET=(0.,0.)
28 EP=(0.,0.)
29 ENT=(0.,0.)
30 ERP=(0.,0.)
31 C!!! CAN SOURCE ILLUMINATE CYLINDER?
32 IF(DTS.LT.-1.5) GO TO 909
33 C!!! COMPUTE PROPAGATION DIRECTION OF RAY BETWEEN
34 C!!! CYLINDER AND PLATE
35 CALL REFIP(PHJR,THJR,PHSR,THSR,MP)
36 SPHJ=SIN(PHJR)
37 CPHJ=COS(PHJR)
38 SJHJ=SIN(THJR)
39 CTHJ=COS(THJR)
40 DJ(1)=CPHJ*STHJ
41 DJ(2)=SPHJ*STHJ
42 DJ(3)=CTHJ
43 AS=PI-THJR
44 SAS=SIN(AS)
45 SASP=ABS(SIN(AS-1.5*PI))
46 CAS=COS(AS)
47 C!!! SPECIFY TANGENT VECTORS
48 TX1=HTS(1)
49 TY1=HTS(2)
50 TX2=HTS(3)
51 TY2=HTS(4)
52 EX(1)=TX1*CPHJ+TY1*SPHJ
53 EX(2)=TX2*CPHJ+TY2*SPHJ
54 C!!! LOOP THRU TANGENT VECTORS
55 I=1
56 LRFST=.FALSE.
57 VI(1)=VTS(1)
58 VI(2)=VTS(2)
59 IF(LDEBUG) WRITE(6,908)
60 908 FORMAT(2,' DEBUGGING SCLRPL SUBROUTINE')
61 CONTINUE
62 CALL NAFUDG(UP,UR,VI(1))
63 SIRA=UN(1)*CPHJ+UN(2)*SPHJ
64 C!!! CALCULATE ALK, THE REFLECTED RAY POL ANGLE IN TAN POINT COORD SYS.
65 ALR=ATAN2(SIRA,-EX(1))
66 IF(ALR.LT.0.) ALR=ALR+PI

```

```

67 C!!! IF CRAZING INCIDENCE IS PRESENT, SKIP TO APPROPRIATE SECTION
68 IF (ABS(CPI-ALR).LT.0.0005) GO TO 5
69 C!!! IF ALR IS G.T. THAN PI, COMPUTE CREEPING WAVE TERMS
70 IF (ALR.GT.PI) GO TO 10
71 C!!! COMPUTE G.O. REFLECTED FIELD TERMS IF ALR .LE. PI
72 CALL HCLRPL(ENT,ERP,MP)
73 C!!! ARE REFLECTED FIELDS PRESENT?
74 IF (LIRFJ) GO TO 1
75 SNAS=UN(1)*XS(1)+UN(2)*XS(2)
76 IC=2*I-1
77 CSAS=BTS(IC)*XS(1)+BTS(IC+1)*XS(2)
78 ALS=BTAN2(SNAS,-CSAS)
79 ALRS=ALR-ALS
80 C!!! IS REFLECTION TO BE HANDLED WITH OTHER TAN VECTOR?
81 IF (ABS(ALRS).LT.0.0005.AND.1.EQ.2) GO TO 1
82 IF (ALRS.LE.-0.0005) GO TO 1
83 GN=(PI*IC)**(1./3.)
84 RNS=RHO1*DL/(DL-RHO1)
85 SKWIG=-ABS(2.*PI*RNS/GN)
86 CF=-SIGN(-2./PI/SKWIG)*CPI**REF
87 CF=CF*EXP(-CJ*SKWIG*SKWIG*SKWIG/12.)
88 TTR=SKWIG/OP
89 AX=PI*(DL+RNS)*TTR*TTR
90 C!!! SPECIFY G.O. REFLECTED FIELD COMPONENTS (FROM REFCYL)
91 EST=ESTH
92 ESP=ESPH
93 ENT=ENTH
94 EHP=EHPH
95 GO TO 30
96 IC
97 IF (LRFST) LRFST(PP)=.FALSE.
98 LRFST=.TRUE.
99 C!!! COMPUTE CREEPING WAVE TERMS IF ALR .GT. PI
100 C!!! COMPUTE INCIDENT RAY TANGENT POINT
101 XI=A*COS(VI(1))
102 YI=B*SIN(VI(1))
103 XD=XI-XS(1)
104 YD=YI-XS(2)
105 S=SQRT(XD*XD+YD*YD)
106 ZI=S*CTHJ/STHJ*XS(3)
107 C!!! IS TANGENT POINT ON CYLINDER?
108 IF (ZI.GT.ZC(1)+XI*CTC(1).OR.
109 2*ZI.LT.ZC(2)+YI*CTC(2)) GO TO 1
110 ZI=ZI-XS(3)
111 PHIR=BTAN2(YD,XD)
112 THIR=BTAN2(S,ZI)
113 S=SQRT(S*S+ZI*ZI)
114 DI(1)=XI/S
115 DI(2)=YD/S
116 DI(3)=ZI/S
117 C!!! DOES INCIDENT RAY HIT PLATE BEFORE CYLINDER?
118 CALL PLRINT(XS,DI,PHIR,P,LHIT)
119 IF (LHIT.AND.(DHIT.LT.S)) GO TO 1
120 C!!! CALCULATE INCIDENT FIELD PATTERN FACTOR
121 CALL SOURCE(EP,EG,FX,FIY,EIZ,THIR,PHIR,VXS)
122 IF (LDERWQ) WRITE(6,*) EP,EG
123 C!!! PERFORM CREEPING WAVE COMPUTATIONS
124 IF (1.EQ.1) VDP=BTAN2(1-R*CO4J,A*SPHJ)
125 IF (1.EQ.2) VDP=BTAN2(1-CPIJ,-A*SPHJ)
126 VDP=VDP-VI(1)
127 IF (VDP.GT.PI) VDP=VDP-PI
128 IF (VDP.LT.-PI) VDP=VDP+PI
129 IF (1.EQ.2) GO TO 21
130 IF (VDP.LT.0.) GO TO 1
131 VL=VI(1)
132 VU=VDP+VI(1)

```

```

133      GO TO 21
134 20 CONTINUE
135      IF(VDP,CT,VV) GO TO 1
136      VL=VDP+V1(1)
137      VU=V1(1)
138 25 CONTINUE
139      CALL FRARG(SK,IG,AS,VL,VU)
140      XRF(1)=A*COS(VD)
141      XRF(2)=L*SIN(VD)
142      LU=3
143      CALL D0G32(VL,VU,FCT,SS)
144      SS=SS/SAS
145      XRF(3)=Z1+SS*CTHJ
146      DO 26 N=1,3
147 26 XRS(N)=XRF(N)
148 C!!! DOES CREEPING WAVE REFLECTION FROM PLATE
149 C!!! MP OCCUR?
150      CALL PLAIN1(XRS,DJ,DHJ,-CP,LHIT)
151      IF(.NOT.LHIT) GO TO 1
152 C!!! IS RAY SHADOWED ANYWHERE?
153      CALL PLAIN1(XRS,D,DH1,MP,LHIT)
154      IF(LHIT) GO TO 1
155      CALL CYLINT(XRS,D,PHSR,DHT,LHIT,.TRUE.)
156      IF(LHIT) GO TO 1
157      CALL PLAIN1(XRF,DJ,DHT,MP,LHIT)
158      IF(LHIT.AND.(DHT.LT.DHJ)) GO TO 1
159      CALL RADCV(RGI,RT,V1(1))
160      CALL RADCV(RGF,RT,VD)
161      GEM=(PI*PI+RGI+RGF)**(1./5.)
162      CF=GEM*CP14*CEXP(-CJ*TP1*(S+SS))/PI/SORT(2.*S)
163 C!!! COMPUTE PHASE TERM
164      CALL IMAGE(XRS,XRF,APR,MP)
165      CF=CF*CEXP(CJ*TP1*(XRS(1)*D(1)+XRS(2)*D(2)+XRS(3)*D(3)))
166      TTRN=SK*IG/GEM
167      AX=PI*S+TTRN*TTRN
168      BX=-UN(2)*D1(3)
169      BY=UN(1)*D1(2)
170      BZ=UN(2)*D1(1)-UN(1)*D1(2)
171      ESP=(B.,B.)
172      EHT=(B.,B.)
173 C!!! COMPUTE HARD AND SOFT CREEPING WAVE COMPONENTS
174      EMP=EIX*UN(1)+E1Y*UN(2)
175      EST=EIX*AX+E1Y*BY+E1Z*BZ
176      IF(1.EQ.1) EMP=-EMP
177      IF(1.EQ.2) EST=-EST
178 28 CONTINUE
179 C!!! COMPUTE THE CYLINDER SCATTERED FIELD
180      XXS=SORT(TP1*XX)
181      AXS=SORT(2.*XX/PI)
182      CALL PHRLS(COC,SSS,XXS)
183      FI=COMPLA(0.5-COC,SSS-0.5)
184      FI=AXS*FI*CEXP(CJ*(1.5*PI+XX))
185      FI=-FI/AXS*IG/SORT(2.)
186      SOTP=SORT(2.*PI)
187      CF=CF*FI*SOTP*CFUN(SX(1))
188      CF=CF*FI*SOTP*CFUN(SX(1))
189      CFPH=CF*EHT*CF*ESP
190      EHT=CFPH*EHT*CF*EST
191      GO TO 6
192 3 CONTINUE
193 C!!! COMPUTE GRAZING INCIDENCE SCATTERED FIELD
194      DO 7 N=1,3
195 7 XRS(N)=XRF(N)
196 C!!! IS REFLECTION FROM PLATE TO OCCUR?
197      CALL PLAIN1(XRS,DJ,DHJ,-P,LHIT)
198      IF(.NOT.LHIT) GO TO 1

```

```

199 C!!! IS RAY SHADOWED ANYWHERE?
200 CALL PLAIN(XS,D,DHT,"P,LHIT)
201 IF(LHIT) GO TO 1
202 CALL CYLINT(XS,D,PHSR,DHT,LHIT,.TRUE.)
203 IF(LHIT) GO TO 1
204 CALL PLAIN(XS,DJ,DHT,"P,LHIT)
205 IF(LHIT.AND.(DHT.LT.DHJT)) GO TO 1
206 SGH=-SIGN(1.,SIN(ALR))
207 C!!! CALCULATE INCIDENT POINT
208 XI=A*COS(VI(1))
209 YI=B*SIN(VI(1))
210 XD=XI-XS(1)
211 YD=YI-XS(2)
212 S=SQRT(XD*XD+YD*YD)
213 ZI=S*CTHJ/STHJ+XS(3)
214 C!!! IS POINT OF GRAZING INCIDENCE OFF OF FINITE CYLINDER?
215 IF(ZI.GT.ZC(1)+XI*CTC(1)).OR.
216 221.LT.ZC(2)+XI*CTC(2)) GO TO 1
217 ZD=ZI-XS(3)
218 S=SQRT(S*S+ZD*ZD)
219 CALL RADCV(RCI,RT,VI(1))
220 GR=(PI*RCI)**(1./3.)
221 C!!! CALCULATE INCIDENT FIELD PATTERN FACTOR
222 CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIN,PHIR,VXS)
223 C!!! CALCULATE PHASE TERM
224 CALL IMAGE(XS,XS,PHIR,IP)
225 CF=CEXP(CJ*PI*(XRS(1)*D(1)+XRS(2)*D(2)+XRS(3)*D(3)))
226 EX=-UN(2)*DJ(3)
227 BY=UN(1)*DJ(3)
228 BZ=UN(2)*DJ(1)-UN(1)*DJ(2)
229 C!!! CALCULATE HARD AND SOFT COMPONENTS OF CYL FIELD
230 EHP=EIX*UN(1)+EIY*UN(2)
231 EST=EIX*EX+EIY*BY+EIZ*BZ
232 IF(1.EQ.1) EHP=-EHP
233 IF(1.EQ.2) EST=-EST
234 CFH=GR*CP1*OPUN(0.)/SQRT(PI*S)
235 CFS=GR*CP1*OPUN(0.)/SQRT(PI*S)
236 C!!! CALCULATE GRAZING INCIDENCE SCATTERED FIELD
237 DETH=(0.5*EF+SGH*CFS*EST)*CF
238 DEPH=(0.5*EG+SGH*CFH*EHP)*CF
239 C CONTINUE
240 C!!! CALCULATE TOTAL CYLINDER FIELDS REFLECTED FROM PLATE
241 VT(1)=VN(UP,2)*D(3)-VN(UP,3)*D(2)
242 VT(2)=VN(UP,3)*D(1)-VN(UP,1)*D(3)
243 VT(3)=VN(UP,1)*D(2)-VN(UP,2)*D(1)
244 C11=VN(UP,1)*CPHJ*CTHJ+VN(UP,2)*SPHJ*CTHJ-VN(UP,3)*STHJ
245 C12=-VN(UP,1)*SPHJ+VN(UP,2)*CPHJ
246 C21=VT(1)*CPHJ*CTHJ+VT(2)*SPHJ*CTHJ-VT(3)*STHJ
247 C22=-VT(1)*SPHJ+VT(2)*CPHJ
248 A1=DETH*C11+DEPH*C12
249 A2=DETH*C21+DEPH*C22
250 C11=VN(UP,1)*DT(1)+VN(UP,2)*DT(2)+VN(UP,3)*DT(3)
251 C12=VN(UP,1)*DP(1)+VN(UP,2)*DP(2)
252 C21=VT(1)*DT(1)+VT(2)*DT(2)+VT(3)*DT(3)
253 C22=VT(1)*DP(1)+VT(2)*DP(2)
254 A3=C11*C22-C12*C21
255 DETH=(A1+C22*A2+C12*A3)/A3
256 DEPH=(A2+C11*A1+C21*A3)/A3
257 EP=EHP+DETH
258 ET=EY+DETH
259 IF(1.EQ.1) WRITE(6,*) 1,SG,IC,X,FI,C
260 IF(1.EQ.1) WRITE(6,*) CPH,CFS
261 IF(1.EQ.1) WRITE(6,*) EMT,EST
262 IF(1.EQ.1) WRITE(6,*) EHP,EPH
263 IF(1.EQ.1) WRITE(6,*) DTH,DEPH
264 1010

```

```

265      IF(I.LE.2) GO TO 3
266 509    CONTINUE
267      IF(.NOT.LTEST) RETURN
268        WRITE(6,510)
269 510    FORMAT(/,' TESTING SCLRPI. SUBROUTINE')
270        WRITE(6,*) ET,EP,JP
271        WRITE(6,*) ERT,ERP
272        RETURN
273        END

```

SCTCYL

PURPOSE

To calculate the far-zone fields scattered by the elliptic cylinder's curved surface.

PERTINENT GEOMETRY

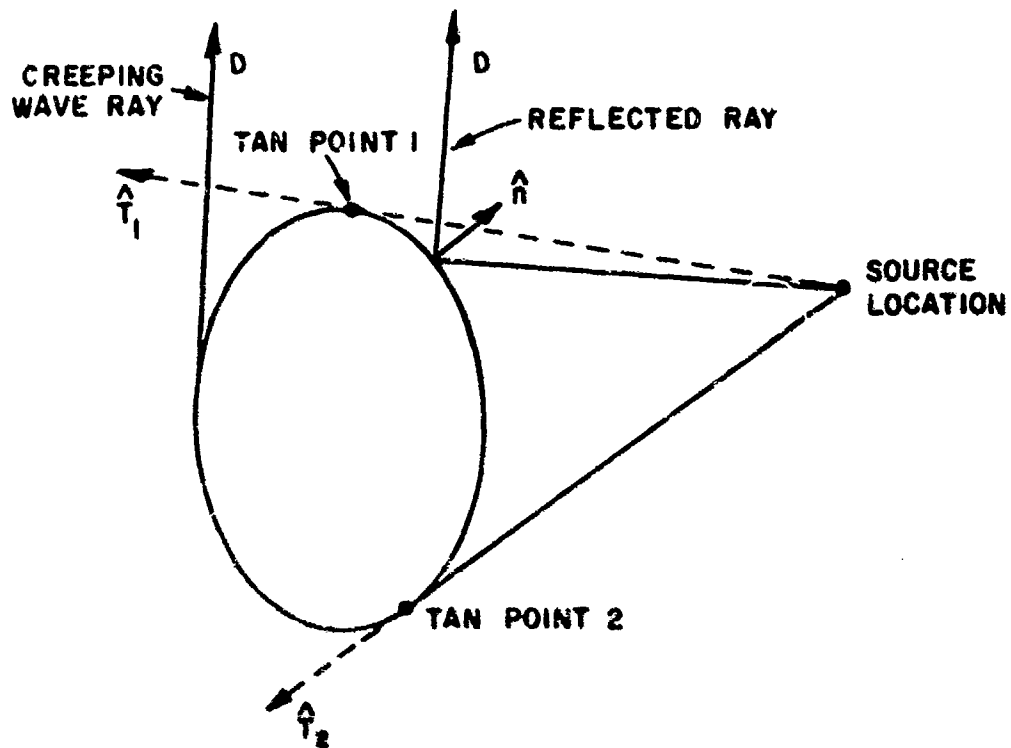


Figure 103--illustration of reflected and creeping wave scattering by the elliptic cylinder.

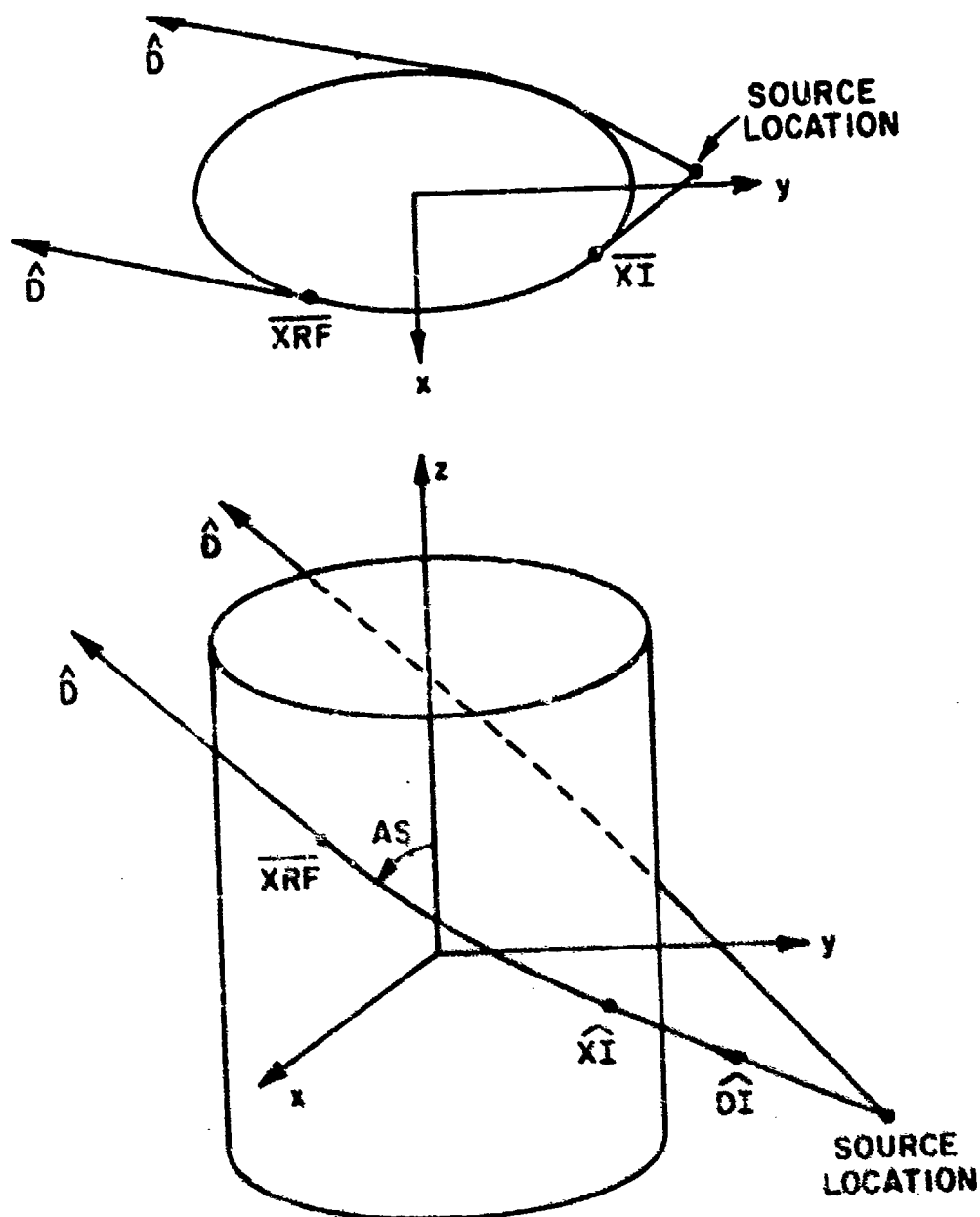


Figure 104--Geometry of creeping wave scattering.

$$\overline{XRF} = \hat{x} \overline{XRF}(1) + \hat{y} \overline{XRF}(2) + \hat{z} \overline{XRF}(3)$$

$$\overline{XI} = \hat{x} \overline{XI}(1) + \hat{y} \overline{XI}(2) + \hat{z} \overline{XI}(3)$$

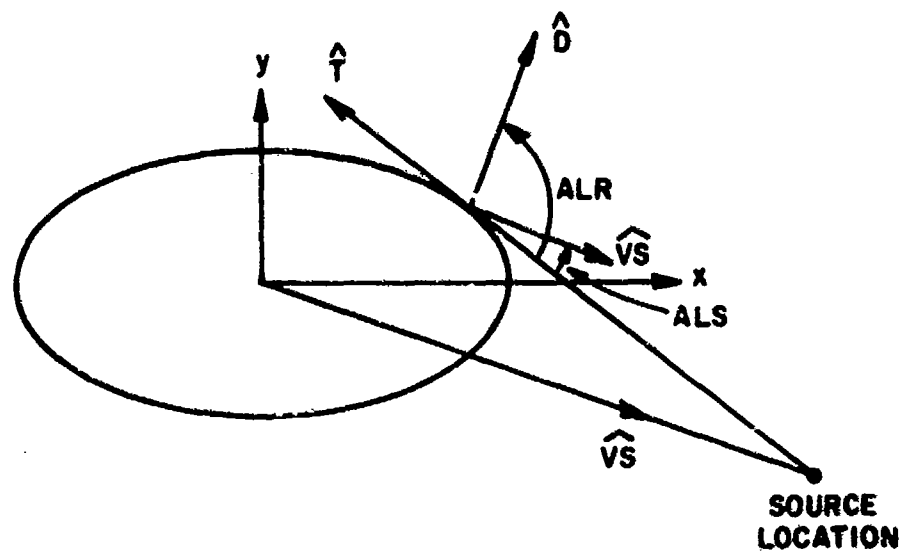


Figure 105--Geometry of angles of cylinder scattering problem.

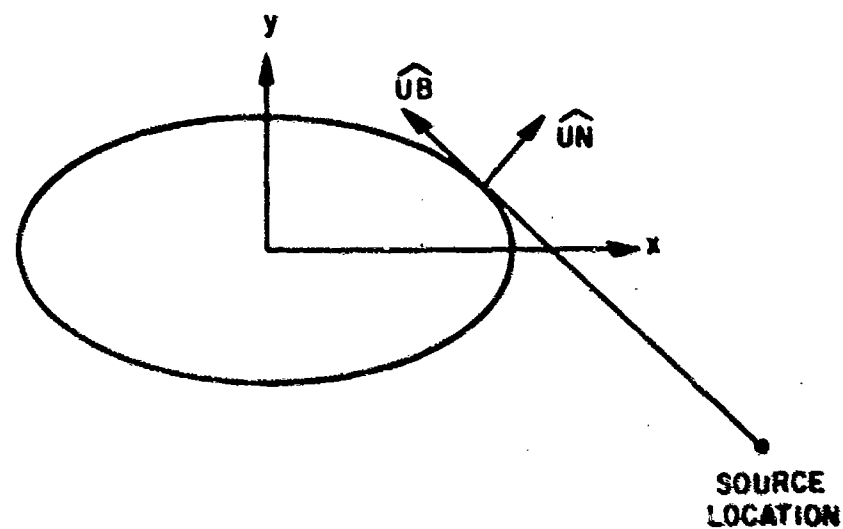


Figure 106--Illustration of tan point coordinate system.

$$\hat{U}_N = \hat{x} U_N(1) + \hat{y} U_N(2)$$

$$\hat{U}_B = \hat{x} U_B(1) + \hat{y} U_B(2)$$

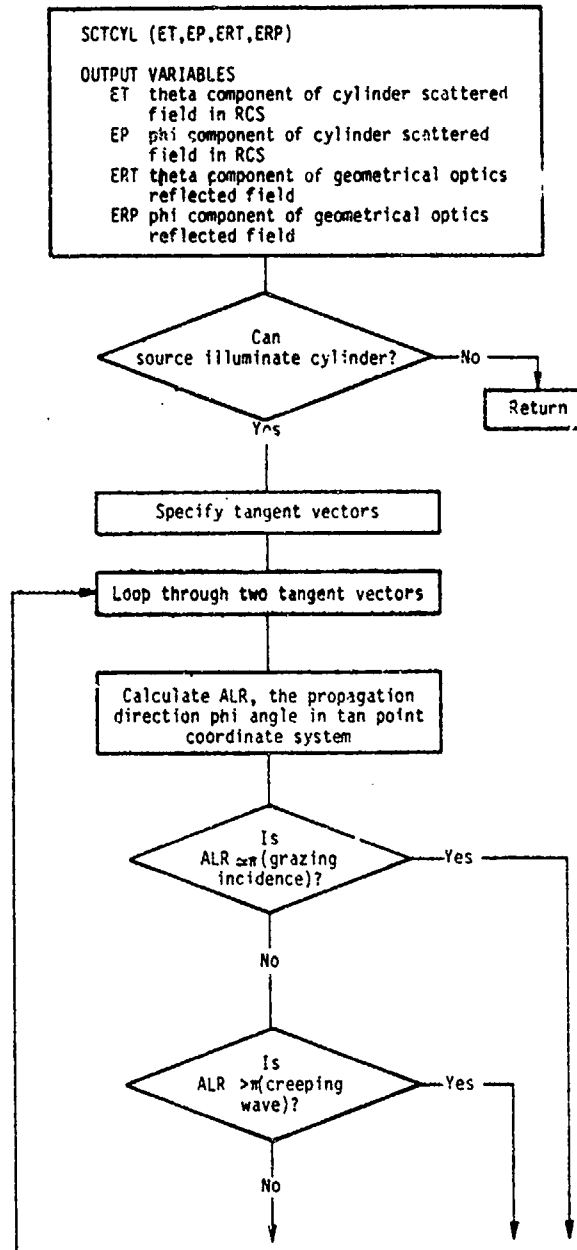
METHOD

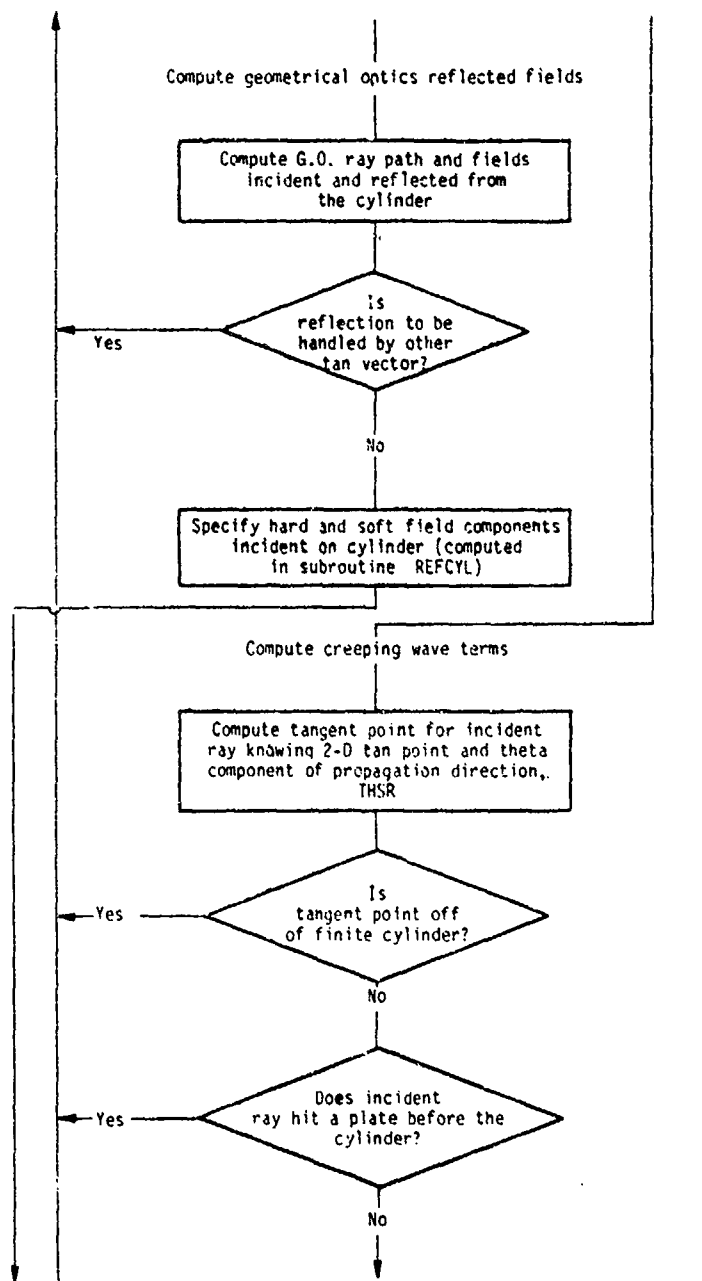
A uniform Geometrical Theory of Diffraction solution[6] is used to compute the reflected and diffracted fields of a source in the presence of the curved surface of an elliptic cylinder. In a given observation direction the solution contains two terms. In the lit region the solution is composed of a reflected field and the dominant creeping wave field, as illustrated in Figure 103. In the shadow region the solution is composed of a clockwise and a counterclockwise creeping wave field, as illustrated in Figure 104. The reflected field and creeping wave fields are modified versions of the usual GTD solution, that is, they are obtained from a uniform solution that is valid at the shadow boundaries (tangent point vector regions) and that goes to the geometrical optics solution in the deep lit region and the usual creeping wave solution in the deep shadow region. The solution is presented in Reference 6 and on pages 112-113 of Reference 1. The phases of the reflected and creeping wave (or transition) fields are referred to the reference coordinate system origin. The fields are combined and the total field scattered by the cylinder is given by

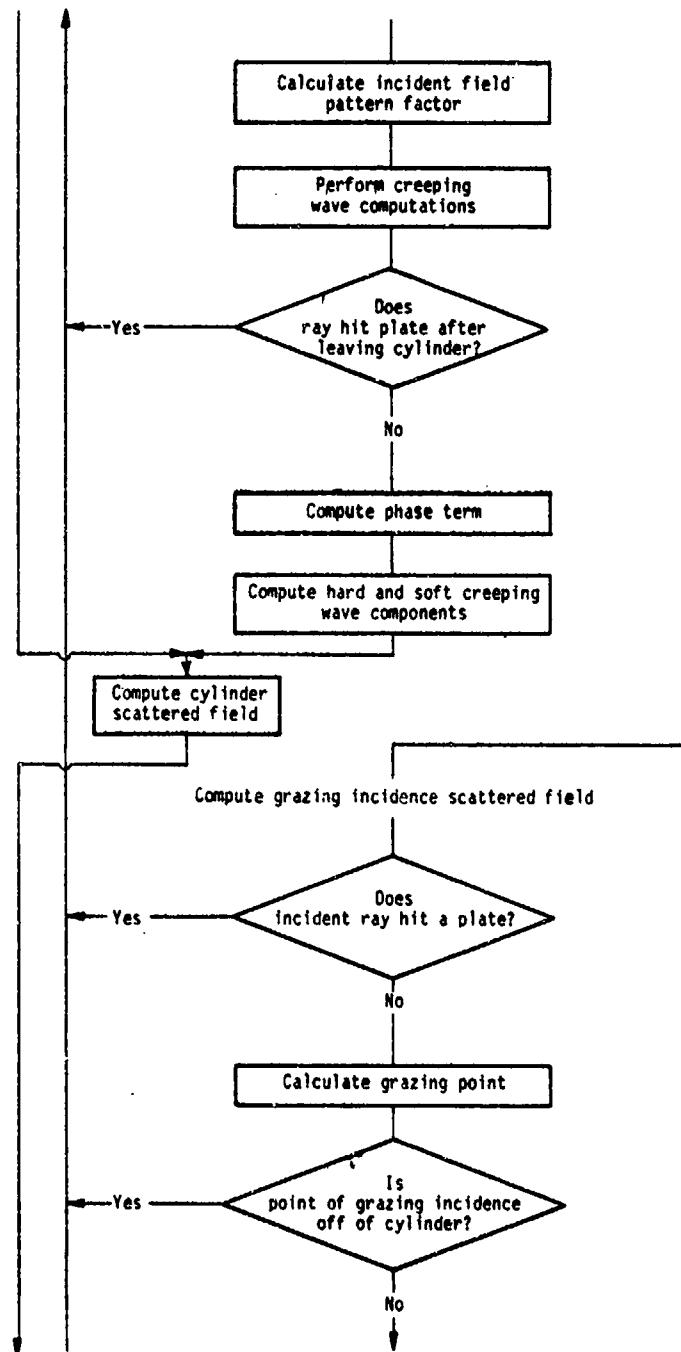
$$\bar{E}^S = W_m (ET\hat{\theta} + EP\hat{\phi}) \frac{e^{-jkR}}{R} ,$$

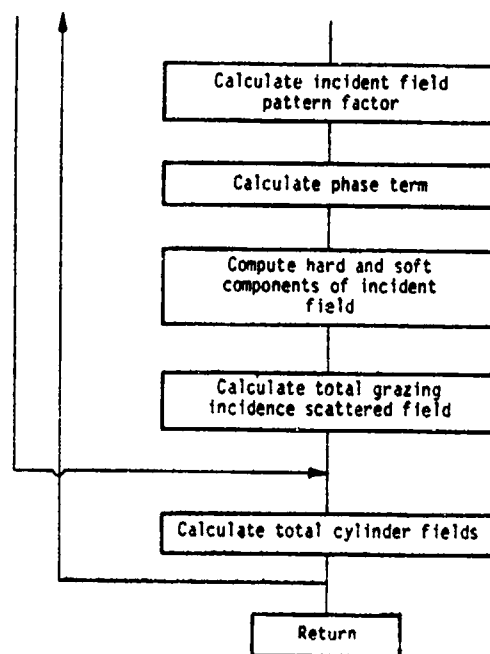
where the factor $\frac{e^{-jkR}}{R}$ and the source weight (W_m) are added elsewhere in the code.

FLOW DIAGRAM









SYMBOL DICTIONARY

ALH PHI ANGLE DEFINING RADIATION DIRECTION IN
TAN POINT COORDINATE SYSTEM (2-D)

ALHS DIFFERENCE BETWEEN ALS AND ALR

ALS PHI ANGLE DEFINING DIRECTION OF RAY FROM RCS
ORI IN TO SOURCE IN TANGENT POINT COORD. SYS

AS ANGLE BETWEEN CREEPING WAVE PATH ON CYL AND LINE
PARALLEL TO Z AXIS

BX } X,Y,Z COMPONENTS OF POLARIZATION UNIT VECTOR
BY } OF SOFT COMPONENT OF FIELD INCIDENT ON CYL (PARALLEL
BZ } TO CYL SURFACE AND NORMAL TO INC RAY PROP DIR)

CF COMPLEX PHASE AND RAY SPREADING COEFFICIENT

CFH HARD TRANSITION FIELD COEFFICIENT

CFS SOFT TRANSITION FIELD COEFFICIENT

CSAS DOT PRODUCT O. CYLINDER TANGENT UNIT VECTOR
AND VECTOR FROM ORIGIN TO SOURCE

D PROPAGATION DIRECTION UNIT VECTOR FOR RAY
SCATTERED FROM CYL IN (X,Y,Z) RCS COMPONENTS

DEPH PHI COMPONENT OF TRANSITION FIELD IN RCS

DETH THETA COMPONENT OF TRANSITION FIELD IN RCS

DHIT DISTANCE FROM SOURCE TO HIT POINT (FROM PLAIN)

DI X,Y,Z COMPONENTS OF UNIT VECTOR OF PROPAGATION
DIRECTION OF RAY INCIDENT ON CYLINDER

EF PATTERN FACTOR FOR THETA COMPONENT OF
INCIDENT FIELD IN RCS

EG PATTERN FACTOR FOR PHI COMPONENT OF
INCIDENT FIELD PATTERN FACTOR IN RCS

EHP PHI COMPONENT OF HARD COMPONENT OF
FIELD INCIDENT ON CYL OR CREEPING WAVE FIELD IN RCS

EHI THETA COMPONENT OF HARD COMPONENT OF
FIELD INCIDENT ON CYL OR CREEPING WAVE FIELD IN RCS

EIX } X,Y,Z COMPONENTS OF INCIDENT FIELD PATTERN FACTOR
EII }
EIZ }

EP PHI COMPONENT OF CYLINDER E FIELD WITH
PHASE REFERRED TO RCS ORIGIN

ER DOT PRODUCT OF UNIT VECTOR TANGENT TO
CYLINDER AND THE PROPAGATION DIR. UNIT VECTOR

ERP PHI COMPONENT OF G.O. REFLECTED FIELD

EKT THETA COMPONENT OF G.O. REFLECTED FIELD

ESP PHI COMPONENT OF SOFT COMPONENT OF
FIELD INCIDENT ON CYL OR CREEPING WAVE FIELD IN RCS

EST THETA COMPONENT OF SOFT COMPONENT OF
FIELD INCIDENT ON CYL OR CREEPING WAVE FIELD IN RCS

ET THETA COMPONENT OF CYLINDER E FIELD WITH
PHASE REFERRED TO RCS ORIGIN

FI PARAMETER USED IN TRANSITION FUNCTION

GM VARIABLE USED IN TRANSITION FUNCTION

I VARIABLE USED TO STEP THROUGH TANGENT POINTS

IC INDEX VARIABLE

LHI1 SET TRUE IF RAY HITS A PLATE (FROM PLAIN)

LTRF (RETURNED FROM RPLRCL) SET TRUE IF G.O.
CYLINDER REFLECTED FIELD DOES NOT EXIST

PHIR PHI COMPONENT OF PROPAGATION DIRECTION OF
RAY INCIDENT ON CYLINDER

RQF RADIUS OF CURV OF CYL AT POINT XRF IN X-Y PLANE

RQI RADIUS OF CURV OF CYL AT INC RAY POINT ON CYL IN XY
PLANE

S LENGTH OF VECTOR FROM SOURCE TO TAN POINT (2 OR 3-D)

SINA DOT PRODUCT OF CYL UNIT NORMAL AND CYL SCATTERED
RAY PROPAGATION DIRECTION UNIT VECTOR

SKNIU PARAMETER USED IN TRANSITION FUNCTION

SNAS DOT PRODUCT OF CYL UNIT NORMAL AND VECTOR FROM
ORIGIN TO SOURCE

THIR THETA COMPONENT OF PROPAGATION DIRECTION
OF RAY INCIDENT ON CYLINDER

TTRM	PARAMETER USED IN TRANSITION FUNCTION
TX1 }	X AND Y COMPONENTS OF UNIT VECTOR OF RAY FROM SOURCE
TY1 }	TANGENT TO TAN POINT 1 OF ELL CYL (2-D)
TX2 }	X AND Y COMPONENTS OF UNIT VECTOR OF RAY FROM SOURCE
TY2 }	TANGENT TO TAN POINT 2 OF ELL CYL (2-D)
UB	X,Y COMPONENTS OF UNIT VECTOR TAN TO CYL AT TAN POINT (2-D)
UN	X,Y COMPONENTS OF UNIT NORMAL TO CYL AT TAN POINT (2-D)
VD	COMPUTATIONAL VARIABLE
VDP	COMPUTATIONAL VARIABLE
VI	ELL. ANGLE USED TO DEFINE TANGENT POINTS (2-D)
VL	ELL ANGLE DEFINING POINT WHERE CREEPING WAVE MEETS CYLINDER
VU	ELL ANGLE DEFINING POINT WHERE CREEPING WAVE LEAVES CYLINDER
XD	X,Y,Z COMPONENTS OF DIRECTION OF RAY FROM
YD	SOURCE TO CYLINDER TANGENT POINT (INCIDENT
ZD	RAY FOR CREEPING AND GRAZING INC. CASES)
XI }	
YI }	X,Y,Z COMPONENTS OF POINT WHERE INCIDENT CREEPING
ZI }	WAVE (OR GRAZING WAVE) MEETS CYLINDER
XPP	X,Y,Z COMPONENTS OF POINT WHERE RAY LEAVES CYLINDER
XRF	X,Y,Z COMPONENTS OF POINT WHERE CREEPING WAVE LEAVES CYLINDER
XX	PARAMETER USED IN TRANSITION FUNCTION
XXS	PARAMETER USED IN TRANSITION FUNCTION
XXX	PARAMETER USED IN TRANSITION FUNCTION

CODE LISTING

```

1 C-----
2 SUBROUTINE SCTCYL(ET,EP,ERT,ERP)
3 C!!!
4 C!!! GTD SCATTERED FIELD OF AN ELLIPTIC CYLINDER
5 C!!!
6 COMPLEX CJ,CPI4,CF,CFH,CFS,FI,PFUN,OFUN
7 COMPLEX EIX,EIY,EIZ,EIPH,EITH,ET,EP,ERT,ERP
8 COMPLEX REF,ESTH,ESPH,EHTH,EHPH,DETH,DEPH,EF,EG
9 COMPLEX EST,ESP,EHT,EHP
10 DIMENSION VI(2),ER(2),UN(2),UB(2),DI(3),XRF(3)
11 LOGICAL LHIT,LTRF,LDEBUG,LTEST,LRFC,LRFCT
12 COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
13 COMMON/SORINF/XS(3),VXS(3,3)
14 COMMON/PIS/PI,TPI,DPR,RPD
15 COMMON/GTD/AS,ID,SAS,SASP,CAS
16 COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
17 COMMON/COMP/CJ,CPI4
18 COMMON/ENDSCL/DTS,VTS(2),BTS(4)
19 COMMON/FUDG/REF,ESTH,ESPH,EHTH,EHPH,XR(3),RG,RHO1,DL,LTRF
20 COMMON/TEST/LDEBUG,LTEST
21 COMMON/CLRFC/LRFC
22 EXTERNAL FCT
23 ET=(0.,0.)
24 EP=(0.,0.)
25 ERT=(0.,0.)
26 ERP=(0.,0.)
27 C!!! CAN SOURCE ILLUMINATE CYLINDER SURFACE?
28 IF(DTS.LT.-1.5) GO TO 909
29 C!!! SPECIFY TANGENT VECTORS
30 TX1=BTS(1)
31 TY1=BTS(2)
32 TX2=BTS(3)
33 TY2=BTS(4)
34 ER(1)=TX1*CPS+TY1*SPS
35 ER(2)=TX2*CPS+TY2*SPS
36 C!!! LOOP THRU TANGENT VECTORS
37 I=1
38 LRFCT=.FALSE.
39 VI(1)=VTS(1)
40 VI(2)=VTS(2)
41 IF(LDEBUG) WRITE(6,900)
42 900 FORMAT(/,' DEBUGGING SCTCYL SUBROUTINE')
43 CONTINUE
44 CALL NANDB(UN,UB,VI(1))
45 SINA=UN(1)*CPS+UN(2)*SPS
46 C!!! CALCULATE ALR, THE PROPAGATION DIRECTION PHI ANGLE
47 C!!! IN TAN POINT COORDINATE SYSTEM.
48 ALR=BTAN2(SINA,-ER(1))
49 IF(ALR.LT.0.) ALR=ALR+TPI
50 C!!! IF GRAZING INCIDENCE IS PRESENT, SKIP TO APPROPRIATE SECTION
51 IF(ABS(PI-ALR).LT.0.0085) GO TO 5
52 C!!! IF ALR IS G.T. THAN PI, COMPUTE CREEPING WAVE TERMS
53 IF(ALR.GT.PI) GO TO 10
54 C!!! IF ALR .LE. PI, COMPUTE G.O. RAY PATH AND FIELD
55 C!!! COMPONENTS
56 CALL HEFCYL(ERT,ERP)
57 IF(LTRF) GO TO 1
58 SNAS=UN(1)*XS(1)+UN(2)*XS(2)
59 IC=2*I-1
60 CSAS=BTS(IC)*XS(1)+BTS(IC+1)*XS(2)
61 ALS=BTAN2(SNAS,-CSAS)
62 ALHS=ALR-ALS
63 C!!! IS REFLECTION TO BG HANDLED WITH OTHER TAN VECTOR?
64 IF(ABS(ALHS).LT.0.0085.AND.I.EQ.2) GO TO 1
65 IF(ALHS.LE.-0.0085) GO TO 1
66 OM=(PI+KG)**(1./3.)

```

```

07      RHS=RHO1*DL/(DL-RHO1)
08      SKWIG=-ABS(2.*PI*RHS/GM/GM)
09      CF=-SORT(-2./PI/SKWIG)*CPI4*REF
70      CF=CF*CEXP(-CJ*SKWIG*SKWIG*SKWIG/12.)
71      TTRM=SKWIG/GM
72      XX=PI*(DL+RHS)*TTRM*TTRM
73 C!!! SPECIFY HARD AND SOFT COMPONENTS OF FIELD
74 C!!! INCIDENT ON CYLINDER (FROM REFCYL)
75      EST=ESTH
76      ESP=ESPH
77      EHT=EHTH
78      EHP=EHPH
79      GO TO 36
80 10    CONTINUE
81      IF(LRFACT) LRFC=.FALSE.
82      LRFACT=.TRUE.
83 C!!! COMPUTE CREEPING WAVE TERMS IF ALR .GT. PI
84 C!!! COMPUTE INCIDENT RAY TANGENT POINT
85      XI=A*COE(VI(1))
86      YI=B*SIN(VI(1))
87      XD=XI-XS(1)
88      YD=YI-XS(2)
89      S=SQRT(XD*XD+YD*YD)
90      ZI=S*CTHS/STHS+XS(3)
91 C!!! IS TANGENT POINT OFF OF FINITE CYLINDER?
92      IF(ZI.GT.ZC(1)+XI*CTC(1).OR.
93      2ZI.LT.ZC(2)+XI*CTC(2)) GO TO 1
94      ZD=ZI-XS(3)
95      PHIR=BTAN2(YD,XD)
96      THIR=BTAN2(S,ZD)
97      S=SQRT(S*S+ZD*ZD)
98      DI(1)=XI/S
99      DI(2)=YD/S
100     DI(3)=ZD/S
101 C!!! DOES INCIDENT RAY HIT PLATE BEFORE CYLINDER?
102     CALL PLAIN(XS,DI,DHIT,D,LHIT)
103     IF(LHIT.AND.(DHIT.LT.S)) GO TO 1
104 C!!! CALCULATE INCIDENT FIELD PATTERN FACTOR
105     CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
106     IF(LDEBUG) WRITE(6,*) EF,EG
107 C!!! PERFORM CREEPING WAVE COMPUTATIONS
108     IF(I.EQ.1) VD=BTAN2(-U*CPS,A*SPS)
109     IF(I.EQ.2) VD=BTAN2(B*CPS,-A*SPS)
110     VDP=VD-VI(1)
111     IF(VDP.GT.PI) VDP=VDP-PI
112     IF(VDP.LT.-PI) VDP=VDP+PI
113     IF(I.EQ.2) GO TO 20
114     IF(VDP.LT.0.) GO TO 1
115     VL=VI(1)
116     VU=VDP+VI(1)
117     GO TO 25
118 20    CONTINUE
119     IF(VDP.GT.0.) GO TO 1
120     VL=VDP+VI(1)
121     VU=VI(1)
122 25    CONTINUE
123     CALL FKARG(SKWIG,AS,VL,VII)
124     XRF(1)=A*COE(VD)
125     XRF(2)=E*SIN(VD)
126     ID=J
127     CALL DCG32(VL,VU,FACT,SS)
128     SS=SS/SA5
129     XRF(3)=ZI*SS*CTHS
130 C!!! DOES RAY HIT PLATE AFTER LEAVING CYLINDER?
131     CALL PLAIN(XRF,D,DHIT,D,LHIT)
132     IF(LHIT) GO TO 1

```

```

133 CALL HALCV(ROI,RT,VI(1))
134 CALL HALCV(RGF,RT,VD)
135 GMM=(PI*PI*ROI*RGF)**(1./6.)
136 CF=-GMM*CP14*CEXP(-CJ*TP1*(5+SS))/PI/SQRT(2.*S)
137 C!!! COMPUTE PHASE TERM
138 CF=CF*CEXP(CJ*TP1*(XRF(1)*D(1)+XRF(2)*D(2)+XRF(3)*D(3)))
139 TTRM=SKWIG/GMM
140 XX=PI*S*TTRM*ITRM
141 BX=-UN(2)*DI(3)
142 BY=UN(1)*DI(3)
143 BZ=UN(2)*DI(1)-UN(1)*DI(2)
144 ESP=(0.,0.)
145 EHT=(0.,0.)
146 C!!! COMPUTE HARD AND SOFT CREEPING WAVE COMPONENTS
147 EHP=EIX*UN(1)+EII*UN(2)
148 EST=EIX*BX+EII*BY+EIZ*BZ
149 IF(1.EQ.1) EFP=-EHP
150 IF(1.EQ.2) EST=-EST
151 GO
152 C!!! COMPUTE THE TRANSITION FIELD
153 XXS=SQRT(TP1*XX)
154 LXX=SQRT(2.*XX/PI)
155 CALL FHNELS(CCC,SSS,XXX)
156 FI=CHPLA(0.5-CCC,SSS-C.5)
157 FI=XXS*FI*CEXP(CJ*(.5*PI+XX))
158 FI=-FI/SKWIG/SQRT(2.)
159 SOTP=SQRT(2.*PI)
160 CFH=CF*(FI+SOTP*QFUN(SKWIG))
161 CFS=CF*(FI+SOTP*PFUN(SKWIG))
162 DEPH=CFH+EHP+CFS*ESP
163 DETH=CFH+EHT+CFS*EST
164 GO TO 6
165 5
166 5
166 C!!! COMPUTE GRAZING INCIDENCE TRANSITION FIELD
167 C!!! DOES RAY HIT PLATE?
168 CALL PLAIN(XS,D,DMIT,0,LHIT)
169 IF(LHIT) GO TO 1
170 SGN=-SIGN(1.,SIN(ALR))
171 C!!! CALCULATE INCIDENT POINT
172 XI=A*COS(VI(1))
173 YI=B*SIN(VI(1))
174 XU=XI-XS(1)
175 YU=YI-XS(2)
176 S=SQRT(XU*XU+YU*YU)
177 ZI=S*CTHS/STHS+XS(3)
178 C!!! IS POINT OF GRAZING INCIDENCE OFF OF FINITE CYLINDER?
179 IF(ZI.LT.ZC(1)+XI*CTC(1).OR.
180 2ZI.LT.ZC(2)+XI*CTC(2)) GO TO 1
181 ZD=ZI-XS(3)
182 S=SQRT(S*S+ZD*ZD)
183 CALL HALCV(ROI,RT,VI(1))
184 GMM=(PI*ROI)**(1./3.)
185 C!!! CALCULATE INCIDENT FIELD PATTERN FACTOR
186 CALL SOURCE(EF,EG,EIX,EIY,EIZ,THSR,PISR,VXS)
187 C!!! CALCULATE PHASE TERM
188 CF=CEXP(CJ*TP1*(XS(1)*D(1)+XS(2)*D(2)+XS(3)*D(3)))
189 BX=-UN(2)*D(3)
190 BY=UN(1)*D(3)
191 BZ=UN(2)*D(1)-UN(1)*D(2)
192 C!!! CALCULATE HARD AND SOFT COMPONENTS OF INCIDENT FIELD
193 EHP=EIX*UN(1)+EII*UN(2)
194 EST=EIX*BX+EII*BY+EIZ*BZ
195 IF(1.EQ.1) EFP=-EHP
196 IF(1.EQ.2) EST=-EST
197 CFH=GMM*CF*(QFUN(0.)/SQRT(PI*S)
198 CFS=GMM*CF*(PFUN(0.)/SQRT(PI*S)

```

```

199 C111  CALCULATE TOTAL GRAZING INCIDENCE FIELD
200      DETH=(0.5*EF*SGN-CFS*EST)*CF
201      DEPH=(0.5*EG*SGN-CFH*ENP)*CF
202 0      CONTINUE
203 C111  CALCULATE TOTAL CYLINDER FIELDS
204      EP=EP+DEPH
205      ET=ET+DETH
206      IF(LDEBUG) WRITE(6,*) I,SK,IG,XX,FI,CF
207      IF(LDEBUG) WRITE(6,*) CFH,CFS
208      IF(LDEBUG) WRITE(6,*) EHT,EST
209      IF(LDEBUG) WRITE(6,*) ENP,ESP
210      IF(LDEBUG) WRITE(6,*) DETH,DEPH
211 1      I=I+1
212      IF(I.LE.2) GO TO 3
213 504    CONTINUE
214      IF(.NOT.LTEST) RETURN
215      WRITE(6,910)
216 910    FORMAT(/,' TESTING SCTCYL SUBROUTINE')
217      WRITE(6,*) ET,EP
218      WRITE(6,*) EHT,ENP
219      RETURN
220      END

```

SOURCE

PURPOSE

To compute the source field pattern factor for radiation in a given direction from the source.

PERTINENT GEOMETRY

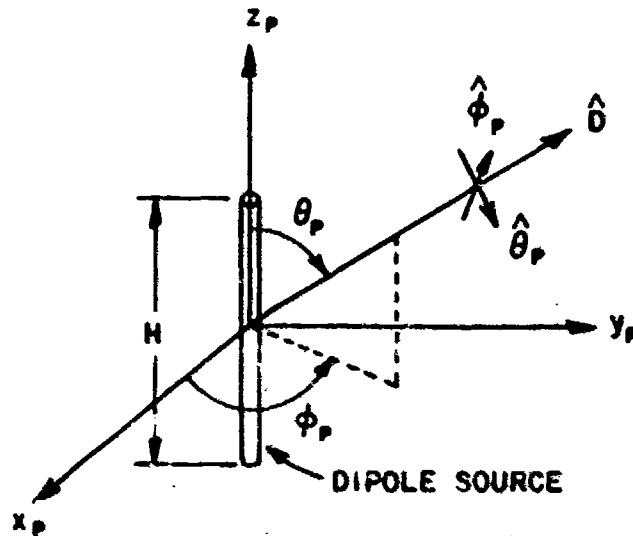


Figure 107--Illustration of one dimensional source (dipole)

Note - one dimensional source always along z_p axis

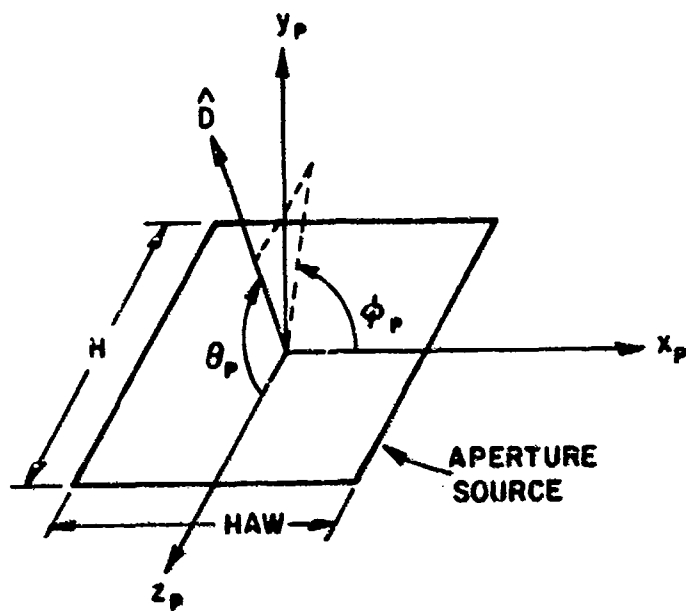


Figure 108--Illustration of two dimensional (aperture) source.

Note - two dimensional source always in x_p - z_p plane with current in the $\hat{z}_p$ direction.

METHOD

The source distribution is given as follows

$$\text{line source: } \left. \begin{matrix} I(z_p) \\ K(z_p) \end{matrix} \right\} = \left\{ \begin{matrix} I_m \\ K_m \end{matrix} \right\} \cos \frac{\pi z_p}{H} \quad x_p=0 \quad y_p=0, \quad \frac{-H}{2} \leq z_p \leq \frac{H}{2}$$

$$\text{aperture source: } \left. \begin{matrix} J(z_p, x_p) \\ M(z_p, x_p) \end{matrix} \right\} = \left\{ \begin{matrix} J_m \\ M_m \end{matrix} \right\} \cos \frac{\pi z_p}{H} \quad y_p=0, \quad \frac{-HAW}{2} \leq x_p \leq \frac{HAW}{2}$$

$$\frac{-H}{2} \leq z_p \leq \frac{H}{2}$$

where x_p, y_p, z_p are unit vectors of the source coordinate systems

$$\begin{aligned} \hat{x}_p &= \hat{x} \text{VAX}(1,1) + \hat{y} \text{VAX}(1,2) + \hat{z} \text{VAX}(1,3) \\ \hat{y}_p &= \hat{x} \text{VAX}(2,1) + \hat{y} \text{VAX}(2,2) + \hat{z} \text{VAX}(2,3) \\ \hat{z}_p &= \hat{x} \text{VAX}(3,1) + \hat{y} \text{VAX}(3,2) + \hat{z} \text{VAX}(3,3). \end{aligned}$$

The far-zone electric field is given by

$$\bar{E}(\theta_p, \phi_p) = \bar{E}_0 F_z(\theta_p) F_x(\theta_p, \phi_p) \frac{e^{-jks'}}{s'}.$$

where for an electric source,

$$\bar{E}_0 = \begin{cases} \hat{\theta}_p \frac{j\eta}{H} I_m H, & \text{line source} \\ \hat{\theta}_p \frac{j\eta}{H} J_m H HAW, & \text{aperture source} \end{cases}$$

and for a magnetic source,

$$\bar{E}_0 = \begin{cases} -\hat{\phi}_p \frac{j}{H} K_m H, & \text{line source} \\ -\hat{\phi}_p \frac{j}{H} M_m H HAW, & \text{aperture source} \end{cases}$$

and where

$$F_z(\theta_p) = \frac{\sin\theta_p \cos(\pi H \cos\theta_p)}{(1-4H^2 \cos^2\theta_p)}$$

$$F_x(\theta_p, \phi_p) = \begin{cases} 1 & , \text{ line source} \\ \frac{\sin(\pi HAW \sin\theta_p \cos\phi_p)}{\pi HAW \sin\theta_p \cos\phi_p} & , \text{ aperture source.} \end{cases}$$

Note that all diagrams and formulae on this and the preceeding page refer to the source coordinate system. The subroutine returns the field components in the reference coordinate system.

The far-zone E-field radiated by the source is then given in the reference coordinate system by

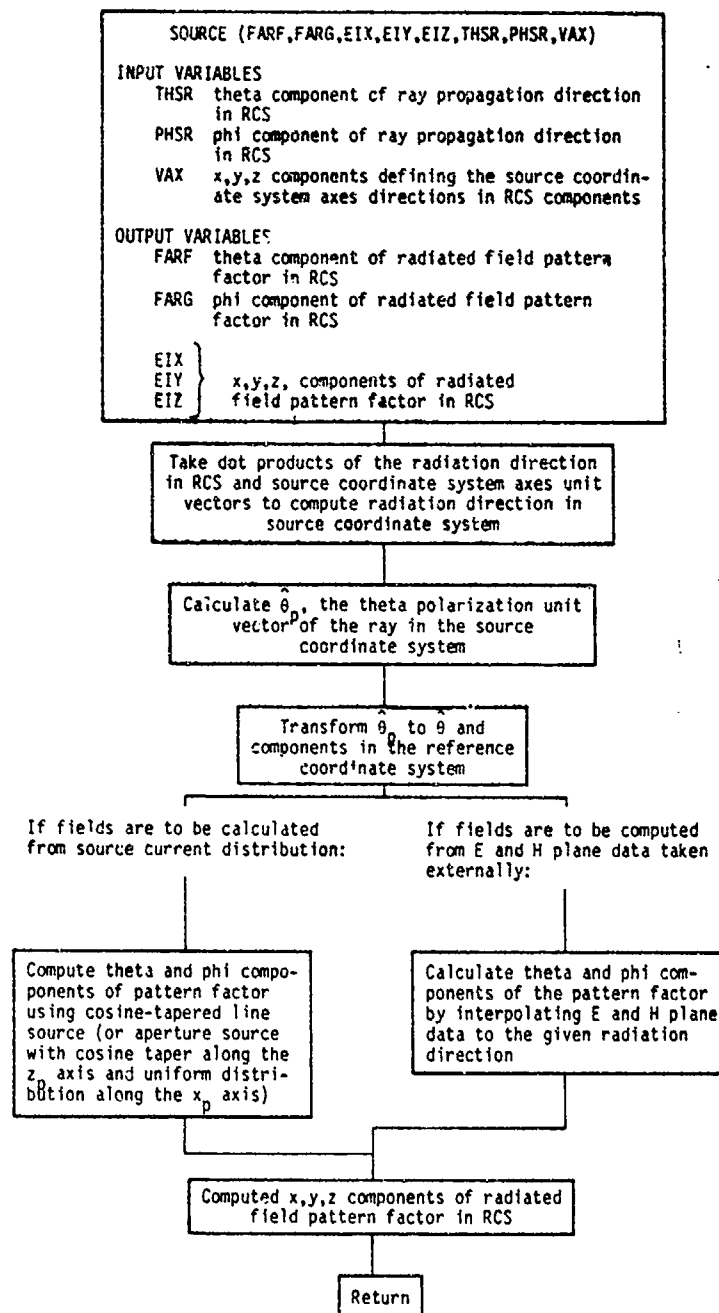
$$\bar{E}(r, \theta, \phi) = W_m (FAR \hat{\theta} + FARG \hat{\phi}) \frac{e^{-jkR}}{R}$$

or

$$\bar{E}(x, y, z) = W_m (EIX \hat{x} + EII \hat{y} + EIZ \hat{z}) \frac{e^{-jkR}}{R}$$

Note that the factor $\frac{e^{-jkR}}{R}$ and the source weights ($W = I, K, J$, or M) are not included in subroutine SOURCE, but are added elsewhere in the code. Note also that the interpolation fields are not fully implemented in this version of the code.

FLOW DIAGRAM



SYMBOL DICTIONARY

AWFAC	PATTERN FACTOR (FX) OF SOURCE FIELD DUE TO XP DIMENSION OF APERTURE
BF	INTERPOLATION VARIABLE
CPHP	COSINE OF PHP
CTHP	COSINE OF THP
EFD	INTERPOLATED E FIELD
EFED	E PLANE SOURCE FIELD PATTERN MEASURED VALUES
EX	COMPUTATIONAL VARIABLE
EXI	PATTERN FACTOR FZ
F	DOT PRODUCT OF THETA UNIT POLARIZATION VECTOR OF SOURCE COORD SYS AND THETA UNIT VECTOR OF RCS
FW	ARGUMENT OF PATTERN FACTOR FX
G	DOT PRODUCT OF THETA UNIT POLARIZATION VECTOR OF SOURCE COORD SYS AND PHI UNIT VECTOR OF RCS
HFD	INTERPOLATED H FIELD
HFED	H PLANE SOURCE PATTERN MEASURED VALUES
IT	INTERPOLATION VARIABLE
PHP	PHI COMPONENT OF RADIATION DIRECTION IN SOURCE COORDINATE SYSTEM
PHSk	PHI COMPONENT OF RADIATION DIRECTION IN RCS
RDX	DOT PRODUCT OF RADIATION DIRECTION AND XP AXIS UNIT VECTOR
RDY	DOT PRODUCT OF RADIATION DIRECTION AND YP AXIS UNIT VECTOR
SPHP	SINE OF PHP
THP	THETA COMPONENT OF RADIATION DIRECTION IN SOURCE COORDINATE SYSTEM
THSR	THETA COMPONENT OF RADIATION DIRECTION IN RCS
VAX	X,Y,Z COMPONENTS DEFINING AXES OF SOURCE (OR SOURCE IMAGE) COORDINATE SYSTEM
XTH	X,Y,Z COMPONENTS OF THE THETA POLARIZATION UNIT VECTOR OF THE RAY IN THE SOURCE COORDINATE SYSTEM (IN RCS COMPONENTS)
YTH	
ZTH	

CODE LISTING

```

1 C-----
2 SUBROUTINE SOURCE(FARF,FARG,EIX,EIY,EIZ,THSR,PHSR,VAX)
3 C!!!
4 C!!! SOURCE FIELD
5 C!!!
6 COMPLEX EX,EIX,EIY,EIZ,FARF,FARG
7 COMPLEX EFED(1),HFED(1),EFD,HFD
8 DIMENSION VAX(3,3)
9 LOGICAL LSOR
10 COMMON/FARP/IM,H,HAW
11 COMMON/PIS/PI,TPI,DPR,RPD
12 COMMON/SOURSF/FACTOR
13 COMMON/FEDDAT/EFED,HFED
14 CTHS=COS(THSR)
15 STHS=SIN(THSR)
16 CPHS=COS(PHSR)
17 SPHS=SIN(PHSR)
18 C!!! TAKE DOT PRODUCTS OF THE RADIATION DIRECTION UNIT
19 C!!! VECTOR AND SOURCE COORD SYS (PRIMED) AXES
20 C!!! UNIT VECTORS TO OBTAIN THP AND PHP (PROPAGATION
21 C!!! ANGLES IN THE SOURCE COORD SYSTEM)
22 CTHP=VAX(3,1)*CPHS*STHS+VAX(3,2)*SPHS*STHS+VAX(3,3)*CTHS
23 RDX=VAX(1,1)*CPHS*STHS+VAX(1,2)*SPHS*STHS+VAX(1,3)*CTHS
24 RDY=VAX(2,1)*CPHS*STHS+VAX(2,2)*SPHS*STHS+VAX(2,3)*CTHS
25 STHP=SQRT(RDX*RDX+RDY*RDY)
26 CPHP=RDY/STHP
27 SPHP=RDY/STHP
28 C!!! CALCULATE THETA POLARIZATION UNIT VECTOR FOR RAY
29 C!!! IN SOURCE COORD SYS AND REPRESENT WITH X,Y,Z
30 C!!! COMPONENTS IN THE REFERENCE COORDINATE SYSTEM
31 XTH=VAX(1,1)*CPHP*CTHP+VAX(2,1)*SPHP*CTHP-VAX(3,1)*STHP
32 YTH=VAX(1,2)*CPHP*CTHP+VAX(2,2)*SPHP*CTHP-VAX(3,2)*STHP
33 ZTH=VAX(1,3)*CPHP*CTHP+VAX(2,3)*SPHP*CTHP-VAX(3,3)*STHP
34 C!!! TRANSFORM THETA POLARIZATION UNIT VECTOR TO
35 C!!! RCS COMPONENTS
36 F=XTH*CTHS*CPHS+YTH*CTHS*SPHS-ZTH*STHS
37 G=-XTH*CPHS+YTH*CPHS
38 IF(IM.EQ.3) GO TO 10
39 C!!! CALCULATE FIELDS USING COSINE TAPERED LINE SOURCE
40 C!!! (OR APERTURE SOURCE WITH COSINE TAPER IN ZP DIRECTION
41 C!!! AND UNIFORM DISTRIBUTION IN THE XP DIRECTION)
42 EX1=STHP
43 ACTHP=ABS(CTHP)
44 IF(ABS(ACTHP)-.5/H).LT.1.E-5) GO TO 5
45 EX1=2.*H*STHP*COS(PI*H*CTHP)/(1.-4.*H*H*CTHP*CTHP)
46 GO TO 6
47 5 EX1=.25*PI*SORT(4.*H*H-1.)
48 6 CONTINUE
49 AWFAC=1.0
50 IF(HAW.LT.0.1) GO TO 7
51 FW=PI*HAW*STHP*CPHP
52 IF(ABS(FW).LT.1.E-05) FW=1.E-05
53 AWFAC=HAW*SIN(FW)/FW
54 7 EX1=EX1*AWFAC
55 EX=CMPLX(0.,EX1*FACTOR)
56 FARF=F*EX*60.
57 FARG=G*EX*60.
58 C!!! USE DUALITY FOR MAGNETIC CURRENT SOURCE
59 IF(IM.EQ.1)FARG=-F*EX/TPI
60 IF(IM.EQ.1)FARF=G*EX/TPI
61 GO TO 20
62 10 CONTINUE
63 C!!! CALCULATE FIELDS BY INTERPOLATION E AND H-PLANE DATA
64 C!!! (TAKEN EXTERNALLY) TO THE GIVEN RADIATION DIRECTION

```

```

05      CTHF=SPHP*STHP
06      BF=CPHP*CPHP*STHP*STHP+CTHP*CTHP
07      STHF=SQRT(BF)
08      THF=DPR*BTAN2(STHF,CTHF)
09      ITF=THF
10      IT=ITF+1
11      EFD=EFED(IT)+(EFED(IT+1)-EFED(IT))*(THF-ITF)
12      HFD=HFED(IT)+(HFED(IT+1)-HFED(IT))*(THF-ITF)
13      IF(ABS(BF).LT.1.E-3) GO TO 15
14      EX=EFD*CPHP*CPHP*STHP*STHP+HFD*CTHP*CTHP
15      EX=EX/BF
16      GO TO 16
17 15    EX=EFD
18 16    CONTINUE
19      FARG=-F*EX
20      FARF=G*EX
21 20    CONTINUE
22 C!!!  COMPUTE X,Y,Z COMPONENTS OF SOURCE PATTERN FACTOR
23      EIX=FARF*CTHS*CPHS-FARG*SPHS
24      EIY=FARF*CTHS*SPHS+FARG*CPHS
25      EIZ=-FARF*STHS
26      RETURN
27      END

```

SOURCE

PURPOSE

To compute the normal derivative, $\frac{\partial \bar{E}^i}{\partial n}$, of the incident field pattern factor for source ray incident on a given edge (to be used in slope diffraction computation).

PERTINENT GEOMETRY

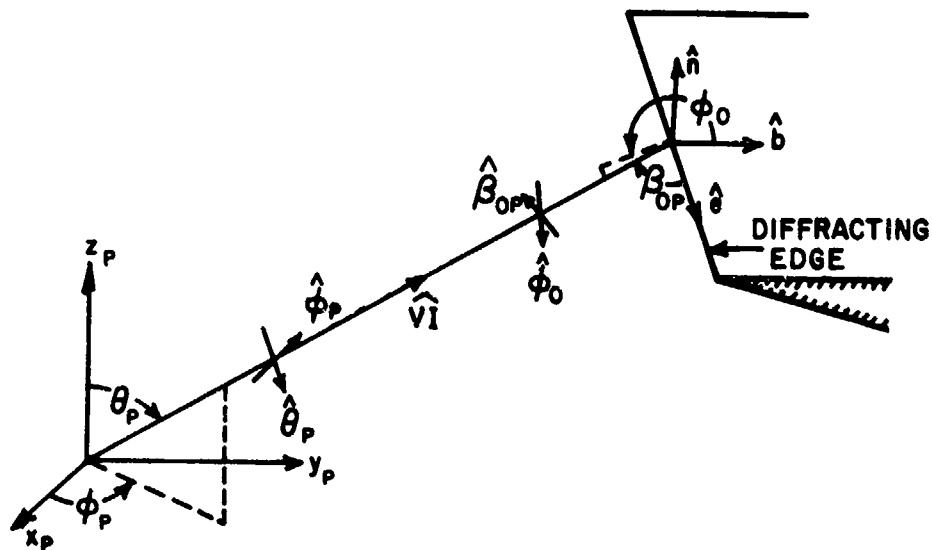


Figure 109--Geometry of source field incident on plate edge.

METHOD

The slope field is given as follows:

$$\frac{\partial}{\partial n} \bar{E}^i = \frac{1}{s' \sin \beta_0} \frac{\partial}{\partial \phi_0} \bar{E}^i$$

where

$$\bar{E}^i = \bar{E}_0 \underbrace{\frac{\sin \theta_p \cos(\pi H \cos \theta_p)}{(1-4H^2 \cos^2 \theta_p)}}_{F_z(\theta_p)} \underbrace{\frac{\sin(\pi HAW \sin \theta_p \cos \phi_p)}{\pi HAW \sin \theta_p \cos \phi_p}}_{F_x(\theta_p, \phi_p)} \frac{e^{-jks'}}{s'}$$

For electric source

$$\bar{E}_0 = \begin{cases} \hat{\theta}_p \frac{j\eta}{\pi} I_m H, & \text{line source} \\ \hat{\theta}_p \frac{j\eta}{\pi} J_m H A W H, & \text{aperture source} \end{cases}$$

$$\bar{E}^i = \bar{E}_0 F_z(\theta_p) F_x(\theta_p, \phi_p) \frac{e^{-jks'}}{s'} = E_{\theta p} \hat{\theta}_p$$

$$\frac{\partial \bar{E}^i}{\partial \phi_0} = \frac{\partial (E_{\theta p} \hat{\theta}_p)}{\partial \phi_0} = \frac{\partial E_{\theta p}}{\partial \phi_0} \hat{\theta}_p + E_{\theta p} \frac{\partial \hat{\theta}_p}{\partial \phi_0}$$

$$\frac{\partial E_{\theta p}}{\partial \phi_0} = \frac{\partial E_{\theta p}}{\partial \theta_p} \frac{\partial \theta_p}{\partial \phi_0} + \frac{\partial E_{\theta p}}{\partial \phi_p} \frac{\partial \phi_p}{\partial \phi_0}$$

$$\frac{\partial E_{\theta p}}{\partial \theta_p} = E_0 \left(\frac{\partial F_z(\theta_p)}{\partial \theta_p} F_x(\theta_p, \phi_p) + F_z \frac{\partial F_x(\theta_p, \phi_p)}{\partial \theta_p} \right) \frac{e^{-jks'}}{s'}$$

$$\frac{\partial E_{\theta p}}{\partial \phi_p} = E_0 \left(F_z(\theta_p) \frac{\partial F_x(\theta_p, \phi_p)}{\partial \phi_p} \right) \frac{e^{-jks'}}{s'}$$

$$F_z(\theta_p) = \frac{\sin \theta_p \cos(\pi H \cos \theta_p)}{(1 - 4H^2 \cos^2 \theta_p)}$$

$$\begin{aligned} \frac{\partial F_z}{\partial \theta_p} = & \{ [(1 - 4H^2 \cos^2 \theta_p) (\cos \theta_p \cos(\pi H \cos \theta_p) + \sin^2 \theta_p \pi H \sin(\pi H \cos \theta_p))] \\ & + [-8H^2 \cos \theta_p \sin^2 \theta_p \cos(\pi H \cos \theta_p)] \} \frac{1}{(1 - 4H^2 \cos^2 \theta_p)^2} \end{aligned}$$

$$F_x = \frac{\sin(\pi \text{ HAW } \sin\theta_p \cos\phi_p)}{\pi \text{ HAW } \sin\theta_p \cos\phi_p}$$

$$\frac{\partial F_x}{\partial \theta_p} = \cot\theta_p \left[\cos(\pi \text{ HAW } \sin\theta_p \cos\phi_p) - \frac{\sin(\pi \text{ HAW } \sin\theta_p \cos\phi_p)}{\pi \text{ HAW } \sin\theta_p \cos\phi_p} \right]$$

$$\frac{\partial F_x}{\partial \phi_p} = \tan\phi_p \left[\frac{\sin(\pi \text{ HAW } \sin\theta_p \cos\phi_p)}{\pi \text{ HAW } \sin\theta_p \cos\phi_p} - \cos(\pi \text{ HAW } \sin\theta_p \cos\phi_p) \right]$$

$$\frac{\partial \theta_p}{\partial \phi_0} = -\sin\beta_{op} \hat{\phi}_0 \cdot \hat{\theta}_p$$

$$\frac{\partial \phi_p}{\partial \phi_0} = -\frac{\sin\beta_{op}}{\sin\theta_p} \hat{\phi}_0 \cdot \hat{\phi}_p$$

$$\frac{\partial \hat{\theta}_p}{\partial \phi_0} = \sin\beta_{op} [\hat{\phi}_0 \cdot \hat{\theta}_p \hat{V}I - \cot\theta_p \hat{\phi}_0 \cdot \hat{\phi}_p \hat{\phi}_p]$$

$$\frac{\partial \hat{\phi}_p}{\partial \phi_0} = \frac{\sin\beta_{op}}{\sin\theta_p} (\hat{\phi}_0 \cdot \hat{\phi}_p) \hat{\rho}_p$$

$$\hat{\rho}_p = \sin\theta_p \hat{V}I + \cos\theta_p \hat{\theta}_p$$

$$\hat{\theta}_p = \hat{x} \text{ XTH} + \hat{y} \text{ YTH} + \hat{z} \text{ ZTH}$$

$$\hat{\phi}_p = \hat{x} \text{ XPH} + \hat{y} \text{ YPH} + \hat{z} \text{ ZPH}$$

combining,

$$\frac{\partial \bar{E}^i}{\partial n} = \frac{j\eta H}{\pi} \left\{ \begin{matrix} I_m \\ J_m \text{ HAW} \end{matrix} \right\} \hat{\phi}_0 \cdot \left[F_x F_z \hat{\theta}_p \hat{V}_I - \left(\frac{\partial F_z}{\partial \theta_p} F_x + F_z \frac{\partial F_x}{\partial \theta_p} \right) \hat{\theta}_p \hat{\theta}_p - \right. \\ \left. \frac{1}{\sin \theta_p} F_z \frac{\partial F_x}{\partial \phi_p} \hat{\phi}_p \hat{\theta}_p - \cot \theta_p F_x F_z \hat{\phi}_p \hat{\phi}_p \right] \frac{e^{-jks'}}{s'^2} .$$

The slope fields for a magnetic source are derived in a similar manner yielding

$$\frac{\partial \bar{E}^i}{\partial n} = \frac{-j}{\pi} H \left\{ \begin{matrix} K_m \\ M_m \text{ HAW} \end{matrix} \right\} \hat{\phi}_0 \cdot \left[F_x F_z \hat{\phi}_p \hat{V}_I - \left(\frac{\partial F_z}{\partial \theta_p} F_x + F_z \frac{\partial F_x}{\partial \theta_p} \right) \hat{\theta}_p \hat{\phi}_p - \right. \\ \left. \frac{1}{\sin \theta_p} F_z \frac{\partial F_x}{\partial \phi_p} \hat{\phi}_p \hat{\phi}_p + \cot \theta_p F_x F_z \hat{\phi}_p \hat{\theta}_p \right] \frac{e^{-jks'}}{s'^2} .$$

The normal derivative of the incident field, $\frac{\partial \bar{E}^i}{\partial n}$, is returned in components perpendicular and parallel to the edge (referred to as hard and soft components):

$$\frac{\partial \bar{E}^i}{\partial n} = \left(\frac{\partial \bar{E}^i}{\partial n} \cdot \hat{\phi}_0 \right) \hat{\phi}_0 + \left(\frac{\partial \bar{E}^i}{\partial n} \cdot \hat{\beta}_{op} \right) \hat{\beta}_{op}$$

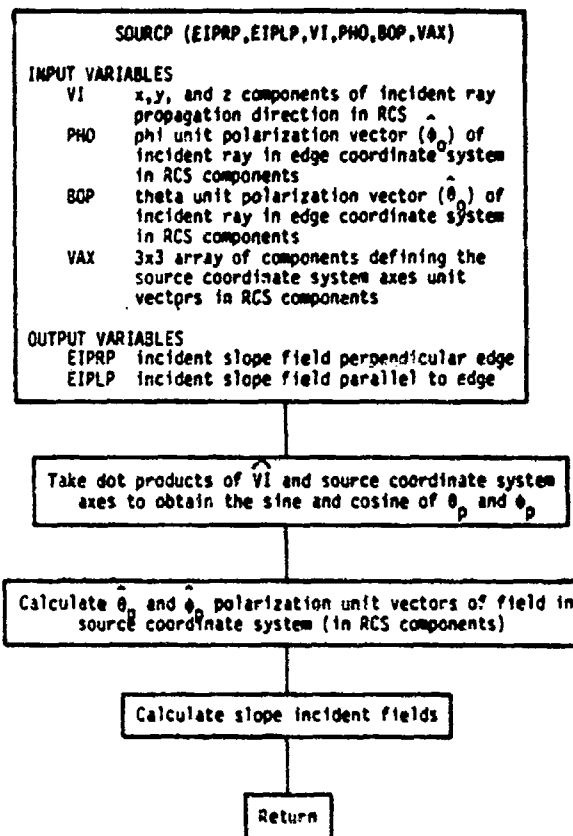
Acoustically
hard case

Acoustically
soft case

$$\frac{\partial \bar{E}^i}{\partial n} = W_m \left[EIPRP \hat{\phi}_0 + EIPLP \hat{\beta}_{op} \right] \frac{e^{-jks'}}{s'^2}$$

Note that the factors $\frac{e^{-jks'}}{s'^2}$, along with the source weights ($W_m = I_m, K_m, J_m$, or M_m) are added elsewhere in the code.

FLOW DIAGRAM



SYMBOL DICTIONARY

ACTHP ABSOLUTE VALUE OF CTHP
 ARG ARGUMENT OF FX
 BOP X,Y,Z COMPONENTS OF BETA POLARIZATION UNIT VECTOR FOR RAY INCIDENT ON EDGE (EDGE-CENTERED COORD SYS)
 CPHP COS(PHP)
 CTHP COS(THP)
 E1 COMPUTATIONAL VARIABLE
 E2 COMPUTATIONAL VARIABLE
 EA COMPUTATIONAL VARIABLE
 EB COMPUTATIONAL VARIABLE
 EFA PARTIAL DERIVATIVE OF FZ WITH THP
 EFB FZ DIVIDED BY SIN(THP)
 EFC PARTIAL OF FX WITH THP DIVIDED BY COT(THP)
 EFD PARTIAL OF FX WITH PHP
 EFE FX TIMES HAW
 EFF PARTIAL OF FP WITH THP
 E1PLP SOFT COMPONENT OF THE SLOPE FIELDS
 E1PHP HARD COMPONENT OF THE SLOPE FIELDS
 PHO X,Y,Z COMPONENTS OF PHI POLARIZATION UNIT VECTOR FOR RAY INCIDENT ON EDGE (EDGE-CENTERED COORD SYS)
 PHP PHI COMPONENT OF PROPAGATION DIRECTION IN SOURCE COORD SYS
 PPBO DOT PRODUCT OF PHI POL UNIT VECTOR OF SOURCE COORD SYS AND BETA POL UNIT VECTOR OF EDGE-CENTERED COORD SYS
 PPHO DOT PRODUCT OF THE PHI POLARIZATION UNIT VECTOR OF THE SOURCE COORD SYS AND THE PHI UNIT POLARIZATION VECTOR OF THE EDGE-CENTERED COORDINATE SYSTEM
 NDA DOT PRODUCT OF VI AND XP AXIS UNIT VECTOR
 NDY DOT PRODUCT OF VI AND YP AXIS UNIT VECTOR
 SN SIGN OF COS(THP)
 SNAHG SIN(ARG)/ARG
 SPHP SIN(PHP)
 STHP SIN(THP)
 THP THETA COMPONENT OF THE PROPAGATION DIRECTION IN THE SOURCE COORDINATE SYSTEM
 TPBO DOT PRODUCT OF THE THETA POLARIZATION UNIT VECTOR OF THE SOURCE COORDINATE SYSTEM AND THE BETA POLARIZATION UNIT VECTOR OF THE EDGE-CENTERED COORDINATE SYSTEM
 TPHO DOT PRODUCT OF THE THETA POLARIZATION UNIT VECTOR OF THE SOURCE COORDINATE SYSTEM AND THE PHI POLARIZATION UNIT VECTOR OF THE EDGE-CENTERED COORDINATE SYSTEM
 VI X,Y,Z COMPONENTS OF THE RAY PROPAGATION DIRECTION IN RCS
 XPH } X,Y,Z COMPONENTS OF THE PHI UNIT POLARIZATION
 YPH } VECTOR OF THE FIELD IN THE SOURCE COORDINATE
 ZPH } SYSTEM IN RCS COMPONENTS
 XTH } X,Y,Z COMPONENTS OF THE THETA UNIT POLARIZATION
 YTH } VECTOR OF THE FIELD IN THE SOURCE COORDINATE
 ZTH } SYSTEM IN RCS COMPONENTS

CODE LISTING

```

1 C-----
2 SUBROUTINE SCURCP(EIPRP,EIPLP,VI,PHO,BOP,VAX)
3 C!!!
4 C!!! INCIDENT SLOPE FIELD
5 C!!!
6 COMPLEX EIPRP,EIPLP
7 DIMENSION VI(3),PHO(3),BOP(3)
8 DIMENSION VAX(3,3)
9 LOGICAL LSOR
10 COMMON/EANP/IN,H,HAN
11 COMMON/PIS/PI,TPI,DPR,RPD
12 COMMON/SCUNSF/FACTOR
13 C!!! TAKE DOT PRODUCTS OF VI AND PRIMED AXES TO OBTAIN THE SINE
14 C!!! AND COSINE OF THP AND PHP
15 HDX=VI(1)*VAX(1,1)+VI(2)*VAX(1,2)+VI(3)*VAX(1,3)
16 HDY=VI(1)*VAX(2,1)+VI(2)*VAX(2,2)+VI(3)*VAX(2,3)
17 CTHP=VI(1)*VAX(3,1)+VI(2)*VAX(3,2)+VI(3)*VAX(3,3)
18 STHP=SQRT(HDX*HDX+HDY*HDY)
19 CPHP=HDX/STHP
20 SPHP=HDY/STHP
21 C!!! CALCULATE THETA AND PHI POL. UNIT VECTORS FOR RAY
22 C!!! IN SOURCE COORD SYS (IN RCS COMPONENTS)
23 XTH=VAX(1,1)*CPHP*CTHP+VAX(2,1)*SPHP*CTHP-VAX(3,1)*STHP
24 YTH=VAX(1,2)*CPHP*CTHP+VAX(2,2)*SPHP*CTHP-VAX(3,2)*STHP
25 ZTH=VAX(1,3)*CPHP*CTHP+VAX(2,3)*SPHP*CTHP-VAX(3,3)*STHP
26 XPH=-SPHP*VAX(1,1)+CPHP*VAX(2,1)
27 YPH=-SPHP*VAX(1,2)+CPHP*VAX(2,2)
28 ZPH=-SPHP*VAX(1,3)+CPHP*VAX(2,3)
29 C!!! CALCULATE SLOPE INCIDENT FIELDS
30 EA=COS(PI*H*CTHP)
31 EB=PI*H*STHP*STHP*SIN(PI*H*CTHP)
32 ACTHP=ABS(CTHP)
33 IF(ABS(ACTHP)-.5/H).LT.1.E-5) GO TO 5
34 E1=1.-4.*H*H*CTHP*CTHP
35 E2=1.-4.*H*H*(2.-CTHP*CTHP)
36 EA=(E2*EA*CTHP/E1+EB)/E1
37 EFB=EA/E1
38 GO TO 6
39 S=SIN(1.,CTHP)
40 EFA=SIN(PI*.4*H*H*PI//16./H
41 EFB=PI/4.
42 C CONTINUE
43 C!!! COMPUTE DOT PRODUCTS OF RAY POLARIZATION UNIT VECTORS
44 C!!! AND UNIT VECTORS PARALLEL AND PERPENDICULAR TO EDGE
45 TPHO=XTH*PHO(1)+YTH*PHO(2)+ZTH*PHO(3)
46 TPHO=XTH*BO(1)+YTH*BO(2)+ZTH*BO(3)
47 PPHO=XPH*PHO(1)+YPH*PHO(2)+ZPH*PHO(3)
48 PPHO=XPH*BO(1)+YPH*BO(2)+ZPH*BO(3)
49 EFO=T.
50 EFC=.
51 EPE=1.
52 IF(INAN.LT.0.1) GOTO 4
53 ARG=PI*HAN*STHP*CPHP
54 IF(ABS(ARG).LT.1.E-05) ARG=1.E-05
55 SHARG=SIN(ARG/ARG)
56 EFC=HAN*(COS(ARG)-SHARG)
57 IF(ABS(CPHP).LT.1.E-05) CPHP=1.E-05
58 EFO=SPHP/CTHP*HAN*(1+ARG-COS(ARG))
59 EFE=HAN*SHARG
60 EFC=EFC*CTHP*EFO+EFC
61 IF(17.00.1) GO TO 10
62 E1=EFO*TPHO;PHO=EFC*17.00*TPHO*CTHP*EFE*SPHP*PHO*PHO
63 E2=EFO*TPHO*TPHO*EFC*17.00*TPHO*CTHP*EFE*EFO*PHO*PHO
64 E(PHO)=2.*PHO*E1*E2*E1*40.*FACTOR

```

```

05      EIPLP=-2.*H*CMPLX(U.,EX2)*60.*FACTOR
06      RETURN
07 10   CONTINUE
08      EX1=EFF*TPHO*PPHO*EFB*EFD*PPHO*PPHO*CTIP*EFE*EFB*PPHO*TPHO
09      EX2=EFF*TPHO*PPHO*EFB*EFD*PPHO*PPHO*CTIP*EFE*EFB*PPHO*TPHO
10      EIPLP=2.*H*CMPLX(U.,EX1)*FACTOR/TP1
11      EIPLP=2.*H*CMPLX(U.,EX2)*FACTOR/TP1
12      RETURN
13      END

```

TANG

PURPOSE

To compute vectors from a source that are tangent to the cylinder in the x-y plane.

PERTINENT GEOMETRY

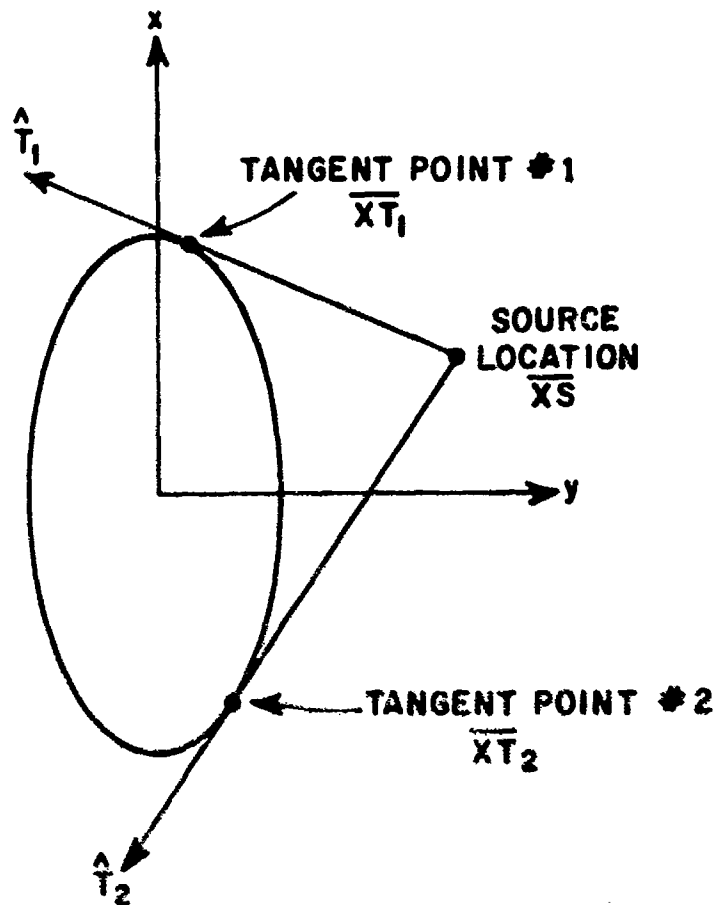


Figure 110--Geometry of source vectors tangent to the cylinder in the x-y plane.

$$\overline{XT_1} = \hat{x} A \cos(VT(1)) + \hat{y} B \sin(VT(1))$$

$$\overline{XT_2} = \hat{x} A \cos(VT(2)) + \hat{y} B \sin(VT(2))$$

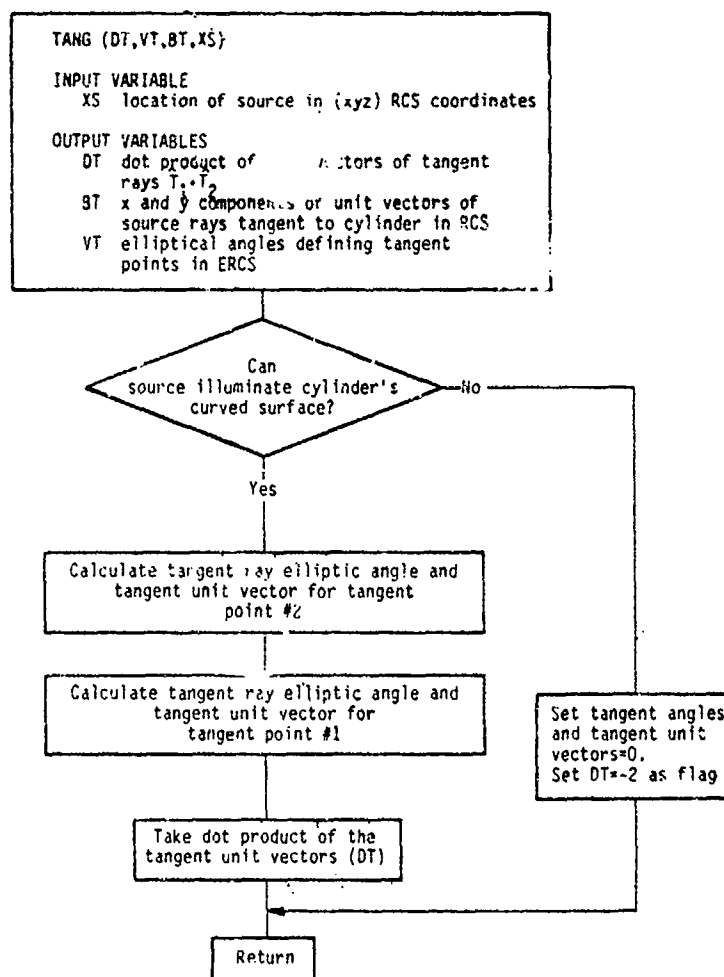
$$\hat{T}_1 = \hat{x} BT(1) + \hat{y} BT(2)$$

$$\hat{T}_2 = \hat{x} BT(3) + \hat{y} BT(4)$$

METHOD

The unit tangent vectors are determined by solving a set of equations found by setting the incident vector from the source equal to the general unit tangent vector to the elliptic surface. Details are given in pages 90-93 in Reference 1.

FLOW DIAGRAM



SYMBOL DICTIONARY

AA DISTANCE FROM SOURCE TO TANGENT POINT
 AL COMPUTATIONAL VARIABLE
 BB DISTANCE FROM ORIGIN TO TANGENT POINT
 BET COMPUTATIONAL VARIABLE
 BT X AND Y COMPONENTS OF TANGENT UNIT VECTORS IN REF COORD SYS.
 CV COSINE OF TANGENT POINT ELL ANGLE
 CVE COSINE OF VE
 DT DOT PRODUCT OF UNIT VECTORS OF THE TWO SOURCE
 RAYS TANGENT TO THE CYLINDER (2-D)
 DV1 ANGLE V1 IN DEGREES
 DV2 ANGLE V2 IN DEGREES
 E1 ERROR DETECTION VARIABLE
 E2 ERROR DETECTION VARIABLE
 RHOE DISTANCE FROM Z AXIS TO POINT WHERE RAY FROM ORIGIN TO
 SOURCE INTERSECTS THE CYLINDER
 RHOS DISTANCE FROM SOURCE TO Z AXIS
 SV SINE OF TANGENT POINT ELL ANGLE
 SVE SINE OF VE
 SX X COMPONENT OF RAY FROM TANGENT POINT TO SOURCE
 SY Y COMPONENT OF RAY FROM TANGENT POINT TO SOURCE
 T1X X COMPONENT OF TANGENT RAY UNIT VECTOR (TAN POINT #2)
 T1Y Y COMPONENT OF TANGENT RAY UNIT VECTOR (TAN POINT #2)
 T2X X COMPONENT OF TANGENT RAY UNIT VECTOR (TAN POINT #1)
 T2Y Y COMPONENT OF TANGENT RAY UNIT VECTOR (TAN POINT #1)
 V1 ELL ANGLE DEFINING TANGENT POINT #2
 V2 ELL ANGLE DEFINING TANGENT POINT #1
 VE ELL ANGLE OF RAY FROM ORIGIN TO SOURCE
 VT ELL ANGLE DEFINING TANGENT POINT LOCATION IN ERCS
 XS SOURCE LOCATION
 XT X-COMPONENT OF TANGENT POINT LOCATION
 YT Y-COMPONENT OF TANGENT POINT LOCATION
 XY COMPUTATIONAL VARIABLE

CODE LISTING

```

1 C-----
2      SUBROUTINE TANG(DT,VT,BT,XS)
3 C!!!
4 C!!!  COMPUTES TANGENT VECTORS TO ELLIPTIC CYLINDER FROM SOURCE
5 C!!!
6      DIMENSION VT(2),BT(4),XS(3)
7      COMMON/PIS/PI,TPI,DPR,RPD
8      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
9      RHOS=SQRT(XS(1)*XS(1)+XS(2)*XS(2))
10 C!!!  CAN SOURCE ILLUMINATE CYLINDER SURFACE?
11      IF(RHOS.GT.A.AND.RHOS.GT.B) GO TO 20
12      IF(RHOS.LT.A.AND.RHOS.LT.B) GO TO 10
13      VE=BTAN2(A*XS(2),B*XS(1))
14      CVE=COS(VE)
15      SVE=SIN(VE)
16      RHOE=SQRT(A*A*CVE*CVE+B*B*SVE*SVE)
17 C!!!  IF SOURCE CANNOT ILLUMINATE CYLINDER, SET ANGLES
18 C!!!  TO ZERO, SET FLAG, AND RETURN
19      IF(RHOS.GE.RHOE) GO TO 20
20 10  CONTINUE
21      DT=-2.
22      VT(1)=0.
23      VT(2)=0.
24      BT(1)=0.
25      BT(2)=0.
26      BT(3)=0.
27      BT(4)=0.
28      RETURN
29 20  CONTINUE
30      XY=B*B*XS(1)*XS(1)+A*A*XS(2)*XS(2)
31      AL=A*A*B*B/XY
32      BET=SQRT(XY-A*A*B*B)/XY
33 C!!!  CALCULATE TAN ANGLE AND TAN UNIT VECTOR FOR TAN POINT #2
34      XT=AL*XS(1)+A*A*BET*XS(2)
35      YT=AL*XS(2)-B*B*BET*XS(1)
36      V1=BTAN2(A*YT,B*XT)
37      CV=COS(V1)
38      SX=XS(1)-A*CV
39      SY=XS(2)-B*SV
40      AA=SQRT(SX*SX+SY*SY)
41      BB=SQRT(A*A*SV*SV+B*B*CV*CV)
42      E1=SQRT((SX/AA+A*SV/BB)**2+(SY/AA-B*CV/BB)**2)
43      T1X=A*SV/BB
44      T1Y=-B*CV/BB
45 C!!!  CALCULATE TAN ANGLE AND TAN UNIT VECTOR FOR TAN POINT #1
46      XT=AL*XS(1)-A*A*BET*XS(2)
47      YT=AL*XS(2)+B*B*BET*XS(1)
48      V2=BTAN2(A*YT,B*XT)
49      SV=SIN(V2)
50      CV=COS(V2)
51      SX=XS(1)-A*CV
52      SY=XS(2)-B*SV
53      AA=SQRT(SX*SX+SY*SY)
54      BB=SQRT(A*A*SV*SV+B*B*CV*CV)
55      E2=SQRT((SX/AA-A*SV/BB)**2+(SY/AA+B*CV/BB)**2)
56      T2X=-A*SV/BB
57      T2Y=B*CV/BB
58 C!!!  TAKE DOT PRODUCT OF TANGENT UNIT VECTORS
59      DT=T1X*T2X+T1Y*T2Y
60      DV1=V1*DPR
61      DV2=V2*DPR
62      VT(1)=V2
63      VT(2)=V1
64      BT(1)=T2X
65      BT(2)=T2Y

```

```
67      BT(3)=TIX
68      BT(4)=TIY
69      IF(E1.GT.1.E-5)WRITE(6,1)DV1,E1
70      IF(E2.GT.1.E-5)WRITE(6,1)DV2,E2
71      FORMAT(1H,'ERROR IN TANGENT SECTION: ',2F10.5)
72      RETURN
73      END
```

CHAPTER V COMMON BLOCK

This chapter defines the variables used in common blocks. The blocks are arranged in alphabetical order.

COMMON BNDUCL -----
 THIS COMMON BLOCK CONTAINS INFORMATION CONCERNING THE STARTING POINT PARAMETERS AND BOUNDS FOR TRACING A RAY DIFFRACTED FROM A PLATE EDGE AND THEN REFLECTED FROM THE CYLINDER. THE INFORMATION IS GENERATED IN SUBROUTINE GEOMPC AND IS USED IN SUBROUTINE DPLRCL.
 VDC(14,6) THIS ARRAY CONTAINS THE ELLIPTIC ANGLE VDC(MP,ME) DEFINING THE STARTING REFLECTION POINT ON THE CYLINDER FOR A RAY DIFFRACTED FROM EDGE ME OF PLATE MP AND THEN REFLECTED BY THE CYLINDER
 UDC(2) THIS ARRAY CONTAINS THE LINEAR VALUE UDC(N) DEFINING THE Z COMPONENT OF THE STARTING REFLECTION POINTS ON THE CYLINDER AXIS. UDC(1) IS FOR THE MOST POSITIVE Z LOCATION AND UDC(2) IS FOR THE MOST NEGATIVE Z LOCATION.
 PDCR(14,6,2) THIS ARRAY CONTAINS ANGLES PDCR(MP,ME,N) DEFINING THE PHI COMPONENT OF THE REFL RAY DIRECTION OF RAYS DIF BY EDGE ME OF PLATE MP AND THEN REFLECTED AT STARTING POINT N ON THE CYLINDER
 TDCR(14,6,2) THIS ARRAY CONTAINS ANGLES TDCR(MP,ME,N) DEFINING THE REFL RAY THETA COMPONENT OF RAY DIRECTIONS FOR RAYS DIF BY EDGE ME OF PLATE MP AND THEN REFLECTED BY STARTING REFLECTION POINT N ON THE CYLINDER.
 DTDC(14,6) DOT PRODUCT OF UNIT VECTORS OF RAYS DIFFRACTED BY EDGE ME OF PLATE MP AND REFLECTED BY THE PREFERRED STARTING POINT ON THE CYLINDER
 BTDC(14,6,4) THIS ARRAY CONTAINS VARIABLES DEFINING THE VECTORS HAVING BEEN DIFFRACTED BY THE CORNER OF EDGE ME OF PLATE MP FURTHEST FROM THE CYLINDER WHICH ARE TANGENT TO THE CYLINDER.
 THE TWO TANGENT VECTORS ARE GIVEN BY:

$$T1 = X * BTDC(MP,ME,1) + Y * BTDC(MP,ME,2)$$

$$T2 = X * BTDC(MP,ME,3) + Y * BTDC(MP,ME,4)$$
 DDC(14,6,2) THIS ARRAY CONTAINS THE COSINE OF THE STARTING REFLECTED RAY THETA ANGLE, WHERE

$$DDC(MP,ME,N) = \cos(TDCR(MP,ME,N))$$

COMMON ENDFCL -----
 THIS COMMON BLOCK IS GENERATED IN SUBROUTINE GEOM AND IS USED TO SPECIFY THE PERMISSABLE RANGE OF DIFFRACTION ANGLES FOR SOURCE RAYS DIFFRACTED BY A PLATE EDGE.
 BD(14,6,2) THIS DEFINES PERMISSABLE THETA DIFFRACTION ANGLES FOR REDGE DIFFRACTION
 THE PERMISSABLE RANGE FOR DIFFRACTION ANGLE B FOR A SOURCE RAY DIFFRACTED BY EDGE ME OF PLATE MP IS GIVEN BY:

$$\cos(B1) < \cos(B0) < \cos(B2)$$
 WHERE B0 IS THE ANGLE THE DIFFRACTED RAY MAKES WITH THE EDGE, AND B1 AND B2 ARE DEFINED AT THE CORNERS OF THE PLATE AS

$$\cos(B1) = BD(MP,ME,1)$$

$$\cos(B2) = BD(MP,ME,2)$$

COMMON BNDICL -----
 THIS COMMON BLOCK CONTAINS INFORMATION RELATED TO VECTORS REFLECTED FROM PLATES WHICH ARE TANGENT TO THE CYLINDER. THE DATA IS GENERATED IN GEOMPC.
 DTI(14) THIS IS THE DOT PRODUCT OF THE TWO RAYS REFLECTED BY PLATE MP WHICH ARE TANGENT TO THE CYLINDER THE CYLINDER FROM THE SOURCE IMAGE FOR REFLECTION FROM PLATE MP:

$$DTI(MP) = T1 \cdot T2$$
 VTI(14,2) THIS IS AN ARRAY OF ELLIPTICAL ANGLES DEFINING THE TWO TANGENT POINTS ON THE CYL FOR RAYS WHICH ARE REFLECTED FROM PLATE MP AND TANGENT TO THE CYLINDER. TANGENT POINT N FOR RAY REFLECTED FROM PLATE MP ARE GIVEN BY:

$X = A * \cos(VTI(MP, N))$
 $Y = B * \sin(VTI(MP, N))$
 B1I(14,4) THIS DEFINES UNIT VECTORS OF THE TWO RAYS REFLECTED
 BY PLATE MP AND TANGENT TO THE CYLINDER.
 THE UNIT VECTOR FOR THE SOURCE RAY REFLECTED FROM
 PLATE MP TANGENT TO TAN POINT 1 IS
 GIVEN BY:
 $T1 = X * B1I(MP, 1) + Y * B1I(MP, 2)$
 THE UNIT VECTOR FOR THE SOURCE RAY REFLECTED FROM
 PLATE MP TANGENT TO TAN POINT 2
 IS GIVEN BY:
 $T2 = X * B1I(MP, 3) + Y * B1I(MP, 4)$

COMMON BNDKCL

THIS COMMON BLOCK CONTAINS INFORMATION CONCERNING THE
 STARTING PARAMETERS AND BOUNDS FOR RAYS REFLECTED FROM THE
 CYLINDER AND THEN DIFFRACTED FROM A PLATE EDGE. THE INFORMATION
 IS GENERATED IN SUBROUTINE GEOMPC AND IS USED IN SUBROUTINE RCLDPL.

VCD(14,6) THIS ARRAY CONTAINS THE ELLIPTIC ANGLE VCD(MP, MC)
 THAT DEFINES THE X, Y COMPONENTS OF THE REFLECTION
 POINT LOCATION FOR THE RAY WHICH IS REFLECTED BY
 THE CYLINDER AND HITS CORNER MC OF PLATE MP.

UCD(14,6) THIS ARRAY CONTAINS THE LINEAR VALUE UCD(MP, MC)
 THAT DEFINES THE Z COMPONENT OF THE REFLECTION POINT
 FOR THE RAY THAT IS REFLECTED BY THE CYLINDER AND
 HITS CORNER MC OF PLATE MP.

THE REFLECTION POINT LOCATION IS GIVEN BY

$$X = A * \cos(VCD(MP, MC))$$

$$Y = B * \cos(VCD(MP, MC))$$

$$Z = UCD(MP, MC)$$

BCD(14,6,2) THIS ARRAY CONTAINS THE VALUE BCD(MP, ME, N)
 THAT DEFINES THE PERMISSABLE RANGE OF THE BETA
 DIFFRACTION ANGLES FOR THE RAY THAT IS REFL BY THE
 CYLINDER AND DIFFRACTED BY EDGE ME OF PLATE MP.
 THE PERMISSABLE RANGE FOR DIFFRACTION ANGLE BO FOR
 THIS RAY IS GIVEN BY:

$$\cos(B1) < \cos(BO) < \cos(B2)$$

WHERE BO IS THE ANGLE THE DIFFRACTED RAY MAKES WITH
 THE EDGE AND ANGLES B1 AND B2 ARE DEFINED AT THE
 CORNERS OF THE PLATE AS:

$$\cos(B1) = BCD(MP, ME, 1)$$

$$\cos(B2) = BCD(MP, ME, 2)$$

COMMON BNDSCCL

THIS COMMON BLOCK CONTAINS INFORMATION RELATED TO VECTORS
 FROM THE SOURCE THAT ARE TANGENT TO THE CYLINDER.

THE DATA IS GENERATED IN SUBROUTINE GEOMC

DTS THIS IS THE DOT PRODUCT OF THE TWO SOURCE VECTORS
 TANGENT TO THE CYLINDER:

$$DTS = T1 \cdot T2$$

VTS(2) VTS CONSISTS OF TWO ELLIPTICAL ANGLES DEFINING THE
 TWO TANGENT POINTS ON THE CYLINDER.

TANGENT POINT N IS GIVEN BY:

$$X = A * \cos(VTS(N))$$

$$Y = B * \sin(VTS(N))$$

BTS(4) THIS DEFINES UNIT VECTORS OF THE TWO SOURCE
 RAYS TANGENT TO THE CYLINDER.

THE UNIT VECTOR FOR THE SOURCE RAY TANGENT TO
TAN POINT 1 IS GIVEN BY:

$$T1 = X * B1S(1) + Y * B1S(2)$$

THE UNIT VECTOR FOR THE SOURCE RAY TANGENT TO
TAN POINT 2 IS GIVEN BY:

$$T2 = X * B1S(3) + Y * B1S(4)$$

COMMON BNDPHN

THIS COMMON BLOCK IS GENERATED IN SUBROUTINE GEO'PC AND IS

USED TO SPECIFY THE BRANCH CUT DISPLACEMENT ANGLE FOR THE
PLATE-CYLINDER REFLECTED-DIFFRACTED AND DIFFRACTED-
REFLECTED TERMS.

PHWR(14,6) IS THE PHI ANGLE LOCATION OF THE CENTER OF EDGE
ME OF PLATE MP, WITH RESPECT TO THE CYLINDER

COMMON CLDRC-----

THIS COMMON BLOCK CONTAINS AN ARRAY OF VARIABLES WHICH
ARE GENERATED IN MAIN AND SUBROUTINE DPLRCL AND ARE
PASSED THROUGH A SUBROUTINE WINDOW TO SUBROUTINE DPRFPT
WHERE THEY ARE USED

LDRC(14,6) IS AN ARRAY OF LOGICAL VARIABLES.

LDRC(MP,ME) IS SET TRUE IF STARTING POINT DATA IS
AVAILABLE FROM PREVIOUS PATTERN ANGLE (FOR NEXT
PATTERN ANGLE) WHEN DEFINING THE REFLECTION POINT ON
CYLINDER FOR A RAY WHICH IS DIFFRACTED FROM EDGE ME OF
PLATE MP AND THEN REFLECTED BY THE CYLINDER

COMMON CLRDC-----

THIS COMMON BLOCK CONTAINS AN ARRAY OF VARIABLES WHICH
ARE GENERATED IN MAIN AND SUBROUTINE RCLDPL AND ARE PASSED
THROUGH A SUBROUTINE WINDOW TO SUBROUTINE RFDFT, WHERE THEY
ARE USED

LRDC(14,6) IS AN ARRAY OF LOGICAL VARIABLES.

LRDC(MP,ME) IS SET TRUE IF STARTING POINT DATA IS
AVAILABLE FROM PREVIOUS PATTERN ANGLE (FOR NEXT
PATTERN ANGLE) WHEN DEFINING THE REFLECTION POINT ON
CYLINDER FOR A RAY WHICH IS REFLECTED BY THE CYLINDER
AND THEN DIFFRACTED BY EDGE ME OF PLATE MP

COMMON CLRFC-----

THIS COMMON BLOCK CONTAINS ONE VARIABLE WHICH IS GENERATED
IN MAIN AND SUBROUTINE REFCYL AND IS PASSED THROUGH
A SUBROUTINE WINDOW TO SUBROUTINE REPTCL, WHERE IT IS USED
LRFC IS A LOGICAL VARIABLE WHICH IS SET TRUE IF THE STARTING
POINT DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE
(FOR NEXT PATTERN ANGLE) WHEN DEFINING THE REFLECTION
POINT ON THE CYLINDER

COMMON CLRFI-----

THIS COMMON BLOCK CONTAINS AN ARRAY OF VARIABLES WHICH ARE
GENERATED IN MAIN AND SUBROUTINE RPLRCL AND ARE PASSED
THROUGH A SUBROUTINE WINDOW TO SUBROUTINE REPTCL, WHERE
THEY ARE USED
LRFI(14) IS AN ARRAY OF LOGICAL VARIABLES. LRFI(MP) IS SET TRUE IF
STARTING POINT DATA IS AVAILABLE FROM PREVIOUS
PATTERN ANGLE (FOR NEXT PATTERN ANGLE) WHEN DEFINING
REFLECTION POINT ON THE CYLINDER FOR A RAY REFLECTED
BY PLATE MP AND THEN REFLECTED BY THE CYLINDER

COMMON CLRFS-----

THIS COMMON BLOCK CONTAINS AN ARRAY OF VARIABLES WHICH IS GENERATED
IN MAIN AND SUBROUTINE RCLRPL AND IS PASSED THROUGH A
SUBROUTINE WINDOW TO SUBROUTINE REPTCL, WHERE IT IS USED.
LRFS(14) IS AN ARRAY OF LOGICAL VARIABLES.

LRFS(MP) IS SET TRUE IF STARTING POINT DATA IS AVAILABLE
FOR THE NEXT PATTERN ANGLE WHEN DEFINING THE REFLECTION
POINT ON A CYLINDER FOR A RAY REFLECTED BY THE
CYLINDER AND THEN REFLECTED BY PLATE MP.

COMMON COMP-----

THIS COMMON BLOCK CONTAINS TWO CONSTANTS USED THROUGHOUT
THE PROGRAM

CJ THE IMAGINARY CONSTANT, J (=SQRT(-1))
CPI4 THE COMPLEX CONSTANT, CEXP(-J*PI/4)

COMMON DIR -----
 THIS COMMON BLOCK CONTAINS INFORMATION SPECIFYING THE DIRECTION OF
 PROPAGATION (THE DESIRED OBSERVATION DIRECTION).
 THE INFORMATION IS COMPUTED IN THE MAIN PROGRAM
 D(3) THE UNIT VECTOR OF THE PROPAGATION DIRECTION IN
 (XYZ) REFERENCE COORDINATE SYSTEM COMPONENTS:
 $D = X \cdot D(1) + Y \cdot D(2) + Z \cdot D(3)$
 THSR THETA ANGLE DEFINING PROPAGATION DIRECTION IN SPHERICAL REFERENCE
 COORDINATE SYSTEM (MEASURED FROM Z-AXIS) IN RADIANS
 PHSR PHI ANGLE DEFINING PROPAGATION DIRECTION IN SPHERICAL REFERENCE
 COORDINATE SYSTEM (MEASURED FROM X-AXIS) IN RADIANS
 SPS THE SINE OF PHSR
 CPS THE COSINE OF PHSR
 STHS THE SINE OF THSR
 CTHS THE COSINE OF THSR

COMMON DOUPEL -----
 THIS COMMON BLOCK CONTAINS INFORMATION DEFINING ANGLES WHERE
 DOUPEL DIFFRACTION TERMS WOULD BE SIGNIFICANT (SHADOW
 BOUNDARIES FOR SINGLE DIFFRACTED RAYS)
 ID(361) THIS INTEGER IDENTIFIES WHICH EDGE THE FIRST
 DIFFRACTION OCCURS FROM AND WHICH PLATE SHADOWS IT
 FOR A GIVEN PATTERN ANGLE, II
 ID(14,0) THIS INTEGER ARRAY IS USED TO STORE THE PLATE THAT
 SHADOWS THE RAY DIFFRACTED FROM EDGE ME OF PLATE
 MP (ID(ME,MP)).
 II THIS INTEGER VARIABLE IDENTIFIES THE OBSERVATION
 ANGLE UNDER CONSIDERATION

COMMON EDMAG -----
 THIS COMMON BLOCK IS GENERATED IN SUBROUTINE GEOM AND IS USED TO
 DEFINE PLATE EDGE LENGTHS
 VMAG(14,0) THIS DEFINES THE LENGTH OF EDGES ON PLATES IN WAVELENGTHS.
 THE LENGTH OF EDGE ME OF PLATE MP IS GIVEN BY
 $VMAG(MP,ME)$

COMMON ESTOR -----
 THIS COMMON BLOCK IS USED IN MAIN TO STORE THE TOTAL ELECTRIC
 FIELDS.
 ETHE(361) THIS COMPLEX ARRAY IS USED TO STORE THE TOTAL
 E-THETA FIELD
 EPHI(361) THIS COMPLEX ARRAY IS USED TO STORE THE TOTAL
 E-PHI FIELD

COMMON FARP -----
 THIS COMMON BLOCK DEFINES THE TYPE OF SOURCE USED AND THE
 DIMENSIONS OF THE SOURCE (VARIABLES DEFINED IN MAIN PROGRAM)
 IM THIS DEFINES THE TYPE OF SOURCE USED:
 IM=0 SPECIFIES ELECTRIC SOURCE
 IM=1 SPECIFIES MAGNETIC SOURCE
 H THE LENGTH OF THE SOURCE (IN THE DIRECTION OF THE SOURCE
 CURRENT) IN WAVELENGTHS
 HAW THE APERTURE WIDTH IN WAVELENGTHS (WIDTH OF THE SOURCE)
 (IF HAW IS LESS THAN 0.1 WAVELENGTHS, THE CODE
 ASSUMES THE SOURCE TO BE A LINE SOURCE)

COMMON FEPAT -----
 THIS COMMON BLOCK CONTAINS SOURCE PATTERN
 FACTOR INFORMATION FOR USE WHEN THE USER
 CHOOSES TO DEFINE THE SOURCE PATTERN FROM DATA OBTAINED ELSEWHERE
 TO BE USED IN AN INTERPOLATION SCHEME
 EFER(361) THIS COMPLEX ARRAY DEFINES THE E-PLANE PATTERN OF THE
 SOURCE
 HFER(361) THIS COMPLEX ARRAY DEFINES THE H-PLANE PATTERN OF THE
 SOURCE

COMMON FHANG -----

THIS COMMON BLOCK DEFINES WEDGE ANGLES FOR PLATE EDGES. IT IS GENERATED IN SUBROUTINE GEOM AND USED IN DIFFRACTION COEFFICIENT CALCULATIONS.

FNP(14,6) WEDGE ANGLE OF EDGE ME OF PLATE MP
 $FNP(MP,ME) = (2 * \pi - WA) / \pi$, WHERE WA IS THE INSIDE ANGLE OF THE WEDGE. IT IS RENAMED FN IN THE MAIN PROGRAM BEFORE CALLING DIFFRACTION SUBROUTINES
 NOTE: IF TWO PLATES INTERSECT, DIFFRACTION CALCULATION IS ONLY CALCULATED ONCE, EVEN THOUGH TWO DIFFERENT EDGES ARE INVOLVED. THEREFORE, THE WEDGE ANGLE FOR ONE OF THE COMMON EDGES WILL BE SET NEGATIVE AS A FLAG AND THE DIFFRACTED FIELD WILL ONLY BE CALCULATED ONCE FOR THE COMMON EDGES (THE FLAGGED EDGE IS IGNORED)

COMMON FUDG -----

THIS COMMON BLOCK IS USED TO TRANSFER DATA CONCERNING GEOMETRICAL OPTICS REFLECTION FROM THE CYLINDER IN SUBROUTINE REFCYL TO SUBROUTINE SCTCYL

TRAN THE SPREAD FACTOR AND PHASE OF THE G.O. FIELD
 ESTH } THE TA AND PHI COMPONENTS OF SOFT COMPONENT OF FIELD INCIDENT
 ESPH } ON CYLINDER REFLECTION POINT
 EHTH } THE TA AND PHI COMPONENTS OF HARD COMPONENT OF FIELD INCIDENT
 ENPH } ON CYLINDER REFLECTION POINT
 XR(3) X,Y,Z COMPONENTS OF THE REFLECTION POINT LOCATION IN RCS
 RG RADIUS OF CURVATURE OF CYLINDER AT REFLECTION POINT
 RHCI RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE AT REFLECTION POINT IN RCS
 SMAG DISTANCE FROM SOURCE TO REFLECTION POINT
 LTRF SET TRUE IF GEOMETRICAL OPTICS REFLECTED FIELD IS NOT PRESENT

COMMON FUDGI -----

THIS COMMON BLOCK IS USED TO TRANSFER DATA CONCERNING GEOMETRICAL OPTICS REFLECTION FROM A PLATE THEN FROM THE CYLINDER IN SUBROUTINE RPLRCL TO SUBROUTINE RPLSCL.

TRAN THE SPREAD FACTOR AND PHASE OF THE GEOMETRICAL OPTICS FIELD
 ESTH THE THETA COMPONENT OF THE SOFT COMPONENT OF THE FIELD INCIDENT ON CYLINDER REFLECTION POINT AFTER PLATE REFLECTION
 ESPH PHI COMPONENT OF SOFT COMPONENT OF THE FIELD INCIDENT ON THE CYLINDER REFLECTION POINT AFTER PLATE REFI.
 EHTH THE TA COMPONENT OF HARD COMPONENT OF FIELD INCIDENT ON CYLINDER REFLECTION POINT AFTER PLATE REFLECTION
 ENPH PHI COMPONENT OF HARD COMPONENT OF FIELD INCIDENT ON CYLINDER REFLECTION POINT AFTER PLATE REFLECTION
 XR(3) X,Y,Z COMPONENTS OF THE REFLECTION POINT LOCATION IN RCS
 RG RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE AT REFLECTION POINT IN RCS
 RHCI RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE AT REFLECTION POINT IN RCS
 SMAG DISTANCE FROM THE SOURCE IMAGE TO THE CYLINDER REFLECTION POINT
 LTRFI SET TRUE IF GEOMETRICAL OPTICS REFLECTED FIELD IS NOT PRESENT.

COMMON FUDJ -----

THIS COMMON BLOCK IS USED TO TRANSFER DATA CONCERNING GEOMETRICAL OPTICS REFLECTION FROM THE CYLINDER AND THEN A PLATE IN SUBROUTINE RCLRPL TO SUBROUTINE SCLRP.

TRAN THE SPREAD FACTOR AND PHASE OF THE G.O. FIELD
 ESTH } THE TA AND PHI COMPONENTS OF SOFT COMPONENT OF FIELD INCIDENT
 ESPH } ON CYLINDER REFLECTION POINT

EHTH THEIA AND PHI COMPONENTS OF HARD COMPONENT OF FIELD INCIDENT
 EHPR ON CYLINDER REFLECTION POINT
 XR(3) X,Y,Z COMPONENTS OF THE REFLECTION POINT LOCATION IN RCS
 RC RADIUS OF CURVATURE OF CYLINDER AT REFLECTION POINT
 RHCI RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE AT
 REFLECTION POINT IN RCS
 SRAC DISTANCE FROM SOURCE TO REFLECTION POINT
 LTRFJ SET TRUE IF GEOMETRICAL OPTICS REFLECTED FIELD
 IS NOT PRESENT

COMMON GEOMEL

THIS COMMON BLOCK CONTAINS INFORMATION DEFINING THE ELLIPTIC CYLINDER GEOMETRY (SPECIFIED IN MAIN PROGRAM FROM DATA INPUT)

A RADIUS OF ELL CYLINDER ALONG X-AXIS OF
 THE CYLINDER IN WAVELENGTHS
 B RADIUS OF ELL CYLINDER ALONG Y-AXIS OF
 THE CYLINDER IN WAVELENGTHS
 ZC(2) POINT WHERE END CAP MC INTERSECTS Z AXIS OF REFERENCE
 COORDINATE SYSTEM
 THE VARIABLE ZC(1) REFERS TO THE MOST POSITIVE
 END CAP AND THE ZC(2) REFERS TO THE MOST NEGATIVE
 END CAP
 SMC(2) THIS IS THE SINE OF THE ANGLE BETWEEN THE Z AXIS AND THE
 PLANE OF END CAP MC (ANGLE MEASURED IN X-Z PLANE)
 CMC(2) THIS IS THE COSINE OF THE ANGLE BETWEEN THE Z AXIS AND THE
 PLANE OF END CAP MC (ANGLE MEASURED IN X-Z PLANE)
 CTC(2) THIS IS THE COTANGENT OF THE ANGLE BETWEEN THE Z AXIS AND THE
 PLANE OF END CAP MC (ANGLE MEASURED IN X-Z PLANE)

COMMON GEOPLA

THIS COMMON BLOCK CONTAINS GEOMETRICAL DATA DEFINING THE GEOMETRY OF THE PLATES (CALCULATED IN SUBROUTINE GEOM)

X(14,6,3) THIS ARRAY DEFINES CORNER LOCATIONS FOR ALL OF THE PLATES IN
 THE (XYZ) REFERENCE COORDINATE SYSTEM COMPONENTS
 IN WAVELENGTHS
 THE LOCATION OF CORNER MC ON PLATE MP IS AS FOLLOWS:
 $X=X(MP,MC,1)$
 $Y=X(MP,MC,2)$
 $Z=X(MP,MC,3)$
 V(14,6,3) THIS DEFINES THE EDGE UNIT VECTOR FOR EACH EDGE ON
 EACH PLATE
 THE EDGE VECTOR V OF EDGE ME ON PLATE MP IS AS FOLLOWS:
 $V=X*V(MP,ME,1)+Y*V(MP,ME,2)+Z*V(MP,ME,3)$
 (NOTE THAT EDGE ME IS BETWEEN CORNERS MC AND MC+1
 WHERE MC=ME)
 VP(14,6,3) THIS DEFINES THE UNIT BINORMAL FOR EACH EDGE ON EACH
 PLATE IN (XYZ) REFERENCE SYSTEM COMPONENTS
 THE EDGE BINORMAL FOR EDGE ME OF PLATE MP IS AS FOLLOWS:
 $VP=X*VP(MP,ME,1)+Y*VP(MP,ME,2)+Z*VP(MP,ME,3)$
 VN(14,3) THIS DEFINES THE UNIT NORMAL FOR EACH PLATE IN (XYZ)
 REFERENCE COORDINATE SYSTEM COMPONENTS
 THE PLATE UNIT NORMAL FOR PLATE MP IS GIVEN AS FOLLOWS:
 $VN=X*VN(MP,1)+Y*VN(MP,2)+Z*VN(MP,3)$
 NEP(14) THIS INTEGER ARRAY DEFINES THE NUMBER OF EDGES
 (OR CORNERS) ON PLATE MP
 NPA THIS INTEGER DEFINES THE NUMBER OF PLATES IN
 THE GEOMETRY (NOT INCLUDING GROUND PLATE)

COMMON GROUND

THIS COMMON BLOCK GIVES INFORMATION CONCERNING THE INFINITE GROUND PLANE

LGND A LOGICAL VARIABLE USED TO INDICATE THE PRESENCE OF AN
 INFINITE GROUND PLANE
 LGND=T INDICATES GROUND PLANE PRESENT
 LGND=F INDICATES GROUND PLANE NOT USED
 MPN THE MAXIMUM NUMBER OF PLATES PRESENT (INCLUDING THE
 GROUND PLANE IF ONE IS USED)

COMMON CTD -----

THIS COMMON BLOCK CONTAINS INFORMATION RELATED TO THE
CREEPING WAVES IN SUBROUTINES SCTCYL, RPLSCL, SCI, L,
FCT, AND RADCV

AS PI MINUS THETA (THETA IS THE THETA COMPONENT OF THE
OBSERVATION DIRECTION IN REFERENCE COORDINATE SYSTEM
RELATIVE TO THE CYLINDER AXIS IN RADIANS)
IDG FLAG FOR FUNCTION FCT
SAS THE SINE OF AS
SASP THE ABSOLUTE VALUE OF THE SINE OF AS-PI/2
CAS THE COSINE OF AS

COMMON HITPLT -----

THIS COMMON BLOCK CONTAINS A VARIABLE THAT IS DEFINED
IN SUBROUTINE PLANT AND IS USED IN SUBROUTINE GEOM
FOR IDENTIFYING DOUBLE DIFFRACTIONS FOR PLATES
MPH THE NUMBER OF THE PLATE WHICH THE RAY HITS FIRST

COMMON IMAINF -----

THIS COMMON BLOCK DEFINES SOURCE IMAGE LOCATIONS AND
DIRECTIONS FOR REFLECTION FROM PLATES. (CALCULATED IN GEOM)
XI(14,14,3) THIS GIVES THE SOURCE IMAGE LOCATIONS IN
WAVELENGTHS FOR ALL SINGLE AND DOUBLE REFLECTIONS
FROM PLATES
THE SOURCE IMAGE LOCATION FOR A RAY WHICH IS SINGLY REFLECTED
FROM PLATE MP IS GIVEN BY:

X=XI(MP,MP,1)
Y=XI(MP,MP,2)
Z=XI(MP,MP,3)

THE SOURCE IMAGE LOCATION FOR A DOUBLY REFLECTED RAY WHICH
REFLECTS OFF OF PLATE MP AND THEN PLATE MPP IS GIVEN BY:

X=XI(MP,MPP,1)
Y=XI(MP,MPP,2)
Z=XI(MP,MPP,3)

VXI(3,3,14) THIS SPECIFIES SINGLE REFLECTION SOURCE IMAGE
COORDINATE SYSTEM AXES UNIT VECTORS IN (XYZ) REFERENCE
COORDINATE SYSTEM COMPONENTS
THE IMAGE SOURCE COORDINATE SYSTEM AXES UNIT VECTORS
FOR SINGLE REFLECTION OF SOURCE IN PLATE MP ARE
GIVEN BY:

GIVEN BY:
X=XI(1,1,MP)+YI(1,2,MP)+ZI(1,3,MP)
Y=XI(2,1,MP)+YI(2,2,MP)+ZI(2,3,MP)
Z=XI(3,1,MP)+YI(3,2,MP)+ZI(3,3,MP)

COMMON IMCINF -----

THIS BLOCK CONTAINS INFORMATION DEFINING THE SOURCE IMAGE
FOR SINGLE REFLECTION FROM A CYLINDER END CAP IN WAVELENGTHS.
THE INFORMATION IS GENERATED IN GEOMC AND IMCPIR.
XIC(2,3) THIS GIVES THE SOURCE IMAGE LOCATIONS FOR SINGLE
REFLECTIONS FROM CYLINDER END CAPS.

THE SOURCE LOCATION FOR REFLECTION FROM
END CAP MC IS GIVEN IN THE MCS AS:

X=XIC(MC,1)
Y=XIC(MC,2)
Z=XIC(MC,3)

VXIC(3,3,2) THIS DEFINES THE SOURCE IMAGE COORDINATE
SYSTEM AXES FOR REFLECTION FROM END CAPS.
THE SOURCE IMAGE COORDINATE SYSTEM AXES UNIT
VECTORS FOR A RAY REFLECTED FROM END CAP MC ARE
GIVEN IN THE MCS AS FOLLOWS:

X=XVXIC(1,1,MC)+YVXIC(1,2,MC)+ZVXIC(1,3,MC)
Y=XVXIC(2,1,MC)+YVXIC(2,2,MC)+ZVXIC(2,3,MC)
Z=XVXIC(3,1,MC)+YVXIC(3,2,MC)+ZVXIC(3,3,MC)

COMMON LDCRY -----
 THIS ARRAY OF VARIABLES IS COMPUTED IN SUBROUTINE GEOMPC
 LDC(14,0) LOGICAL VARIABLE
 LDC(MP,ME) IS SET TRUE IF EDGE ME OF PLATE MP IS
 PART OF A DIFFRACTING WEDGE USED TO COMPUTE
 DIFFRACTED FIELDS FOR PLATE DIFFRACTED, CYLINDER
 REFLECTED RAY

COMMON LOGDIF -----
 THIS COMMON BLOCK CONTAINS INFORMATION THAT INDICATES WHETHER OR
 NOT SLOPE AND CORNER DIFFRACTION
 MECHANISMS ARE TO BE INCLUDED IN FIELD CALCULATIONS
 LSLOPE A LOGICAL VARIABLE USED TO INDICATE IF SLOPE DIFFRACTION
 IS DESIRED
 LSLOPE=T INDICATES SLOPE DIFFRACTION DESIRED
 LSLOPE=F INDICATES SLOPE DIFFRACTION NOT DESIRED
 LCORNR A LOGICAL VARIABLE USED TO INDICATE IF CORNER DIFFRACTION
 IS DESIRED
 LCORNR=T INDICATES CORNER DIFFRACTION DESIRED
 LCORNR=F INDICATES CORNER DIFFRACTION NOT DESIRED

COMMON LPLCY -----
 THIS COMMON BLOCK CONTAINS LOGICAL VARIABLES INDICATING THE PRESENCE
 OR ABSENCE OF PLATES AND CYLINDERS IN THE GEOMETRY (SPECIFIED IN
 MAIN PROGRAM)
 LPLA A LOGICAL VARIABLE USED TO INDICATE THE PRESENCE OF AT
 LEAST ONE PLATE OR INFINITE GROUND PLATE
 LPLA=T INDICATES PLATES ARE PRESENT
 LPLA=F INDICATES PLATES NOT PRESENT
 LCYL A LOGICAL VARIABLE USED TO INDICATE THE PRESENCE OF
 AN ELLIPTIC CYLINDER
 LCYL=T INDICATES CYLINDER PRESENT
 LCYL=F INDICATES CYLINDER NOT PRESENT

COMMON LSHDP -----
 THIS COMMON BLOCK IS USED TO TRANSFER DATA BETWEEN SUBROUTINE GEOM
 AND SUBROUTINE PLAINF FOR THE TOTAL SHADOWING ALGORITHM
 LSTS A LOGICAL VARIABLE SET TRUE IF TOTAL SHADOWING ALGORITHM
 IS BEING USED
 LSTD(14) A LOGICAL ARRAY SUCH THAT
 LSTD(ML) IS SET TRUE IF PLATE ML TOTALLY SHADOWS PLATE MP
 FROM THE SOURCE

COMMON LSHET -----
 THIS COMMON BLOCK CONTAINS INFORMATION INDICATING PLATES THAT
 ARE TOTALLY SHADOWED FROM THE SOURCE OR PLATES WHICH ARE SHADOWED
 FROM OTHER PLATES (GENERATED IN SUB. GEOM AND USED IN MAIN PROGRAM)
 LSHD(14) A LOGICAL VARIABLE USED TO INDICATE IF PLATE MP IS TOTALLY
 SHADOWED FROM THE SOURCE BY ANY ONE PLATE OR THE CYLINDER
 LSHD(MP)=T INDICATES PLATE MP IS TOTALLY SHADOWED FROM
 DIRECT SOURCE RAYS
 LSHD(MP)=F INDICATES PLATE MP IS NOT TOTALLY SHADOWED
 LIHD(14,14) A LOGICAL VARIABLE USED TO INDICATE IF PLATES
 MP AND MPP CANNOT ILLUMINATE EACH OTHER
 LIHD(MP,MPP)=T INDICATES PLATES CANNOT ILLUMINATE EACH OTHER
 LIHD(MP,MPP)=F INDICATES PLATES CAN ILLUMINATE EACH OTHER

COMMON OUTPD -----
 THIS COMMON BLOCK CONTAINS INFORMATION USED TO OBTAIN THE
 PROPER FIELD OUTPUT IN SUBROUTINE OUTPUT.
 LPRAD THIS LOGICAL VARIABLE IS SET TRUE IF TOTAL POWER
 RADIATED BY THE SOURCES IS SPECIFIED BY THE USER
 LRAZD THIS LOGICAL VARIABLE IS SET TRUE IF COMPUTED
 FAR-ZONE FIELD VALUES ARE TO INCLUDE RANGE FACTOR
 ($\exp(-j\pi r)/r$)
 PRAD TOTAL POWER RADIATED (OR INPUT POWER) IN WATTS
 (SPECIFIED BY THE USER)

RALD THE DISTANCE FROM THE ORIGIN TO THE FAR FIELD
 POINT IN METERS
 WL THE WAVELENGTH IN METERS

COMMON PATAI

THIS COMMON BLOCK DEFINES THE PATTERN CUT COORDINATE SYSTEM.
 XPC(3) THIS DEFINES THE PATTERN CUT COORD SYS X AXIS UNIT
 VECTOR IN (XYZ) REF. COORD. SYS. COMPONENTS
 THE X AXIS UNIT VECTOR IS GIVEN AS:

$$\hat{X}PC = \hat{X} * XPC(1) + \hat{Y} * XPC(2) + \hat{Z} * XPC(3)$$

 YPC(3) THIS DEFINES THE PATTERN CUT COORD SYS Y AXIS UNIT
 VECTOR IN (XYZ) RCS COMPONENTS
 THE Y AXIS UNIT VECTOR IS GIVEN AS:

$$\hat{Y}PC = \hat{X} * YPC(1) + \hat{Y} * YPC(2) + \hat{Z} * YPC(3)$$

 ZPC(3) THIS DEFINES THE PATTERN CUT COORD SYS Z AXIS UNIT
 VECTOR IN (XYZ) REF. COORD. SYS. COMPONENTS
 THE Z AXIS UNIT VECTOR IS GIVEN AS:

$$\hat{Z}PC = \hat{X} * ZPC(1) + \hat{Y} * ZPC(2) + \hat{Z} * ZPC(3)$$

COMMON PIS

THIS COMMON BLOCK CONTAINS MATHEMATICAL CONSTANTS BASED ON
 THE NUMBER, PI WHICH ARE USED THROUGHOUT THE PROGRAM
 THEY ARE DEFINED IN THE BLOCK DATA.
 PI THE CONSTANT, PI (3.14159265)
 TPI A CONSTANT, TWO TIMES PI (6.28318531)
 DPR THE CONVERSION FACTOR FOR CONVERTING ANGULAR MEASUREMENTS
 IN RADIANS TO DEGREES (=180/PI=57.2957795)
 RPD THE CONVERSION FACTOR FOR CONVERTING ANGULAR MEASUREMENTS IN
 DEGREES TO RADIANS (=PI/180=0.0174532925)

COMMON ROTRDT

THIS COMMON BLOCK DEFINES THE NEW REFERENCE COORDINATE SYSTEM AXES
 DIRECTIONS. IT IS DEFINED FROM INPUT DATA IN THE MAIN
 PROGRAM AND IS USED IN SUBROUTINE ROTRAN TO TRANSFORM LOCATIONS
 AND VECTORS FROM OLD REF COORD SYSTEM COMPONENTS TO NEW REFERENCE
 COORDINATE SYSTEM COMPONENTS. THE NEW REFERENCE COORDINATE SYSTEM IS
 THE CYLINDER COORDINATE SYSTEM (IF A CYLINDER IS PRESENT).
 IF THE CYLINDER IS NOT PRESENT THE TRANSFORMATION IS NOT NECESSARY
 SINCE THE REFERENCE COORDINATE SYSTEM REMAINS THE SAME COORDINATE
 SYSTEM IN WHICH THE GEOMETRY WAS DEFINED
 XCL(3) THIS DEFINES THE NEW REFERENCE COORDINATE SYSTEM X-AXIS UNIT
 VECTOR IN OLD REFERENCE SYSTEM COMPONENTS
 THE RCS X-AXIS UNIT VECTOR IS DEFINED AS:

$$\hat{X} = \hat{X}O * XCL(1) + \hat{Y}O * XCL(2) + \hat{Z}O * XCL(3)$$

 YCL(3) THIS DEFINES THE NEW REFERENCE COORDINATE SYSTEM Y-AXIS UNIT
 VECTOR IN OLD REFERENCE SYSTEM COMPONENTS
 THE RCS Y-AXIS UNIT VECTOR IS DEFINED AS:

$$\hat{Y} = \hat{X}O * YCL(1) + \hat{Y}O * YCL(2) + \hat{Z}O * YCL(3)$$

 ZCL(3) THIS DEFINES THE NEW REFERENCE COORDINATE SYSTEM Z-AXIS UNIT
 VECTOR IN OLD REFERENCE SYSTEM COMPONENTS
 THE RCS Z-AXIS UNIT VECTOR IS DEFINED AS:

$$\hat{Z} = \hat{X}O * ZCL(1) + \hat{Y}O * ZCL(2) + \hat{Z}O * ZCL(3)$$

 WHERE $\hat{X}O, \hat{Y}O, \hat{Z}O$ ARE UNIT VECTORS OF THE OLD REFERENCE
 COORD SYS AXES

COMMON SOURCE

THIS COMMON BLOCK CONTAINS INFORMATION PERTAINING TO THE LOCATION
 AND ORIENTATION OF THE SOURCE UNDER CONSIDERATION (SPECIFIED IN
 MAIN PROGRAM)
 XS(3) THE LOCATION OF THE SOURCE IN (XYZ) REFERENCE COORDINATE
 SYSTEM COMPONENTS IN WAVELENGTHS
 VXS(3,3) A 3X3 MATRIX DEFINING THE SOURCE COORDINATE
 SYSTEM AXES UNIT VECTORS IN REFERENCE COORDINATE SYSTEM

COMPONENTS:

$$\begin{aligned} X_P &= X \cdot VXS(1,1) + Y \cdot VXS(1,2) + Z \cdot VXS(1,3) \\ Y_P &= X \cdot VXS(2,1) + Y \cdot VXS(2,2) + Z \cdot VXS(2,3) \\ Z_P &= X \cdot VXS(3,1) + Y \cdot VXS(3,2) + Z \cdot VXS(3,3) \end{aligned}$$

COMMON SOURCE

THIS COMMON BLOCK CONTAINS A SOURCE FIELD FACTOR.
IT IS COMPUTED IN SUBROUTINES GEOM AND GEOMC AND IS USED
IN SUBROUTINE SOURCE AND SOURCEP.
FACTOR THIS IS A COEFFICIENT OF THE SOURCE FIELD USED
TO OBTAIN THE CORRECT FIELD MAGNITUDE FOR SOURCES
MOUNTED ON PLATES OR END CAPS (IN ORDER TO
COMPENSATE FOR IMAGE EFFECTS). FACTOR IS GIVEN AS
FOLLOWS:

FOR ELECTRIC SOURCES:

FOR SOURCE NOT MOUNTED ON PLATE OR END CAP,
FACTOR=1.0
FOR SOURCE MOUNTED NORMAL TO PLATE OR END CAP,
FACTOR=1.0
FOR SOURCE MOUNTED ON PLATE OR END CAP BUT NOT
NORMAL TO IT,
FACTOR=0.5

FOR MAGNETIC SOURCES:

FOR SOURCE NOT MOUNTED ON PLATE OR END CAP,
FACTOR=1.0
FOR SOURCE MOUNTED ON PLATE OR END CAP AND
PARALLEL TO IT,
FACTOR=2.0
FOR SOURCE MOUNTED ON PLATE OR END CAP, BUT NOT
PARALLEL TO IT,
FACTOR=1.0

COMMON SRFACC

THIS COMMON BLOCK IS DEFINED IN SUBROUTINE GEOMC AND IS USED IN THE
MAIN PROGRAM

LSRPF(MC) A LOGICAL VARIABLE INDICATING WHETHER OR NOT
THE SOURCE UNDER CONSIDERATION IS MOUNTED ON CYLINDER
END CAP MC
LSRPF(MC)=T INDICATES SOURCE MOUNTED ON END CAP MC
LSRPF(MC)=F INDICATES SOURCE NOT MOUNTED ON END CAP MC

COMMON SURFAC

THIS BLOCK IS DEFINED IN SUBROUTINE GEOM AND IS USED IN THE MAIN
PROGRAM AND IN SEVERAL SUBROUTINES

LSURF(MP) A LOGICAL VARIABLE INDICATING WHETHER OR NOT
THE SOURCE UNDER CONSIDERATION IS MOUNTED ON PLATE MP
LSURF(MP)=T INDICATES SOURCE MOUNTED ON PLATE MP
LSURF(MP)=F INDICATES SOURCE NOT MOUNTED ON PLATE MP

COMMON TEST

THIS COMMON BLOCK CONTAINS LOGICAL VARIABLES USED TO INSTRUCT
THE CODE WHETHER OR NOT A PRINT-OUT OF TEST DATA IS DESIRED.
LDEBUG THIS LOGICAL VARIABLE IS SET TRUE IF DEBUGGING DATA IS TO
BE PRINTED ON LINE PRINTER
LTEST THIS LOGICAL VARIABLE IS SET TRUE IF TEST DATA IS TO
BE PRINTED ON LINE PRINTER

COMMON THPHUV

THIS COMMON BLOCK CONTAINS INFORMATION DEFINING THE THETA AND PHI
POLARIZATION UNIT VECTORS FOR THE OBSERVATION DIRECTION (COMPUTED IN
MAIN PROGRAM)

DT(3) THE THETA UNIT VECTOR FOR OBSERVATION DIRECTION D
IN RCS COMPONENTS:

$$DT = X \cdot DT(1) + Y \cdot DT(2) + Z \cdot DT(3)$$

DP(2) THE PHI UNIT VECTOR FOR OBSERVATION DIRECTION D IN REFERENCE

COORDINATE SYSTEM COMPONENTS:
 $DP = X * DP(1) + Y * DP(2) + Z * DP(3)$

COMMON TOPD -----
THIS COMMON BLOCK CONTAINS A CONSTANT USED IN THE DIFFRACTION
SUBROUTINES
TOP THE COMPLEX CONSTANT, $-CEXP(-J * PI / 4)$

CHAPTER VI SYSTEM LIBRARY FUNCTIONS USED BY CODE

ACOS(X) = arccos of X; result in radians
 AINT(X) = X truncated to an integer and set real
 ALOG10(X) = log to base ten of X
 ATAN2(Y,X) = arctangent of Y/X; result in radians covering all four quadrants
 CABS(Z) = magnitude of the complex number, Z
 CEXP(Z) = complex exponential (e^Z)
 CLOG(Z) = complex log of Z ($\ln Z + j \tan^{-1} \frac{\text{Im}(Z)}{\text{Re}(Z)}$)
 CONJG(X) = complex conjugate of Z
 COS(X) = cosine of X
 CSQRT(Z) = square root of a complex number, Z
 SIGN(X,Y) = sign of Y times absolute value of X
 SIN(X) = sine of X
 SQRT(X) = square root of X
 TAN(X) = tangent of X

REFERENCES

1. R.J. Marhefka, "Analysis of Aircraft Wing-Mounted Antenna Patterns," Report 2902-25, June 1976, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Grant No. NGL 36-008-138 for National Aeronautics and Space Administration.
2. W.D. Burnside, M.C. Gilreath, R.J. Marhefka, and C.L. Yu, "A Study of KC-135 Aircraft Antenna Patterns," IEEE Trans. on Antennas and Propagation, Vol. AP-23, No. 3, May 1975, pp. 309-316.
3. C.L. Yu, W.D. Burnside and M.C. Gilreath, "Volumetric Pattern Analysis of Airborne Antennas," IEEE Trans. on Antennas and Propagation, Vol. AP-26, No. 5, Sept. 1978, pp. 636-641.
4. R.G. Kouyoumjian and P.H. Pathak, "A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly-Conducting Surface," Proc. IEEE, Vol. 62, November 1974, pp. 1448-1461.
5. R.G. Kouyoumjian, "The Geometrical Theory of Diffraction and Its Applications," Numerical and Asymptotic Techniques in Electromagnetics, edited by R. Mittra, Springer-Verlag, New York, 1975.
6. P.H. Pathak, W.D. Burnside and R.J. Marhefka, "A Uniform GTD Analysis of the Diffraction of Electromagnetic Waves by a Smooth Convex Surface," submitted for publication to IEEE Trans. on Antennas and Propagation. (Also Report 784583-4, April 1979, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Contract No. N62269-76-C-0554 for Naval Air Development Center.
7. G.J. Burke and A.J. Poggio, "Numerical Electromagnetic Code (NEC) - Method of Moments," NOSC/TD 116, July 1977, Naval Ocean Systems Center, San Diego, California 92152.
8. R.J. Marhefka and W.D. Burnside, "NEC - Basic Scattering Code User's Manual," Report 784508-18, in progress, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Contract No. N00123-76-C-1371 for Naval Regional Procurement Office.
9. W.D. Burnside and P.H. Pathak, "A Corner Diffraction Coefficient," to appear.
10. Y.M. Hwang and R.G. Kouyoumjian, "A Dyadic Diffraction Coefficient for an Electromagnetic Wave which is Rapidly Varying at an Edge," USNC-URSI 1974 Annual Meeting, Boulder, CO., Oct. 1974.

11. IBM, System 1360 Scientific Subroutine Package, (360A-CM-03Y),
Version III, Programmer's Manual, IBM Corp., 1968, p. 303.
12. E.D. Greer and W.D. Burnside, "High Frequency Near Field
Scattering by an Elliptic Disk," Report 4583-1, December 1976,
The Ohio State University ElectroScience Laboratory, Department
of Electrical Engineering; prepared under Contract No. H62269-
76-C-0554 for Naval Air Development Center.
13. J. Boersma, "Computation of Fresnel Integrals," "Math Comp.,
Vol. 14, 1960, p. 380.
14. P.H. Pathak, "An Asymptotic Analysis of the Scattering of Plane
Waves by a Smooth Convex Cylinder," paper to appear in J. Radio
Science. (Also The Ohio State University ElectroScience Laboratory
Tech. Report 784583-3, March 1978).
15. N.A. Logan, "General Research in Diffraction Theory," Vol. I, LMSD-
288087; and Vol. II, LMSD-288088, Missiles and Space Division,
Lockheed Aircraft Corp., 1959.
16. A. Ralston, A First Course in Numerical Analysis, McGraw-Hill Book
Co., New York, 1965, pp. 371-373.